Non-Split Balancing Higher Order Tree for multi-privileged groups

A.MUTHULAKSHMI
Department of Mathematics
and Computer Applications
PSG College of Technology
Coimbatore
India
muthulakshmi.a@gmail.com

R.ANITHA
Department of Mathematics
and Computer Applications
PSG College of Technology
Coimbatore
India
anitha_nadarajan@mail.psgtech.ac.in

M.SUMATHI
Department of Mathematics
and Computer Applications
PSG College of Technology
Coimbatore
India
ramasumathi.psg@gmail.com

Abstract: *In group communication scenario, key management is important to preserve forward and backward secrecy. In order to achieve it keys need to be changed during user join/leave which is done by an operation called rekeying. In a centralized key management scheme, the server thereafter passes the new keys to the existing users who are affected via unicasts and multicasts. The number of unicasts and multicasts decide the rekeying cost. B-trees and NSBHO trees help to reduce the rekeying cost as compared to the binary trees. This paper discusses the use of these trees in Multi-privileged environment providing the algorithms for user join/leave as well as rekeying algorithms in both the cases. The bounds for the heights of the trees have been given and also the rekeying costs in the three cases have been addressed.*

Keywords-Multi-privileged group; rekeying; balanced tree;B-tree, NSBHO tree.

1. Introduction

Applications like video conferencing, pay-per-view channels, distance learning, distribution of stock quotes and news, need transmission of a single message to multiple recipients. In these applications, if, the same data is unicasted to every individual user, huge amount network bandwidth is needed, bringing down the server efficiency. It can be improved by multicasting the data to all the users simultaneously thereby saving the network bandwidth considerably. Access control is an important aspect in this scenario.

Access control mechanisms must be deployed to achieve a successful group-oriented multicast. A group key is the key which allows a group of users to decrypt a broadcast message that is intended for that group and not others. Key tree approach is a hierarchy of keys where each member is assigned a set of keys based on its location in the key tree.

A group key has to be updated and redistributed or calculated safely when there is a change with membership. The newly joined users should not be able to derive the previous group keys and the revoked users should not be able to derive the future session keys with previously distributed keying information.

Binary trees have been used for key management in Logical Key Hierarchy approach. Due to the increase in the number of users B-trees were used. Even with B-trees, during user join there is need for node splitting which results in many multicasts for rekeying and this can be avoided with Non-Split Balancing Higher Order Trees which does not need node splitting with user join.

In conventional group communication scheme, all members in a group have same level of access privileges. But many group applications have multiple related data streams and members have different access privileges. Multi- Group key management scheme was proposed for such a case. There are two challenging factors involved in this scheme. Users can subscribe to one or more data streams which are encrypted by separate Session Keys. The challenge inherent here is to ensure that the users cannot access beyond their privileges. The second factor is to provide a flexible group key management scheme when the users join/change /leave their group.

This paper proposes a scheme using Non-Split Balancing High-Order (NSBHO) tree for multi-privileged scenario. It also shows that this scheme is efficient in reducing the number of multicast messages during user join/leave while using NSBHO as compared to the case of height balanced 2-3 trees. The rest of the paper is organized as follows.

Section 2 details the preliminaries, related work and also provides applications of the proposed work, while section 3 explains the proposed work with

This work is a part of the Collaborative Directed Basic Research on Smart and Secure Environment project funded by NTRO, India

algorithms and examples. Section 4 is a discussion of results of the proposed scheme. Section 5 gives the conclusion and future work.

2. Preliminaries

Group communication applications that need copies of data uses Internet Protocol multicast and the group key is distributed by the key server. Scalable rekeying for Internet protocol multicast was proposed by Bhattacharjee[1].

2.1 Key tree approach

Key tree is one of the approaches to manage keys, wherein each user is assigned a set of keys based on its location in the tree. A group key has to be updated and redistributed or calculated safely when there is a change with membership. The newly joined users should not be able to derive the previous group keys, even if they are able to derive future group keys with subsequently distributed keying information. Similarly, the revoked users should not be able to derive the future session keys, even if they are able to compute the previous session keys with previously distributed keying information. Rekeying is the mechanism that changes the affected old keys to ensure forward and backward secrecy in a key tree. It is illustrated in fig.1, where the keys gk, sk_1, sk_3 are affected when user u_3 leaves and gk', sk_1', sk_3' are affected when u_3 joins. The affected keys are changed after a join/leave. The number of messages that need to be distributed to the members to let them obtain the new group key is the rekeying cost. When users join or leave the key tree, the rekeying cost increases with logarithm of group size.

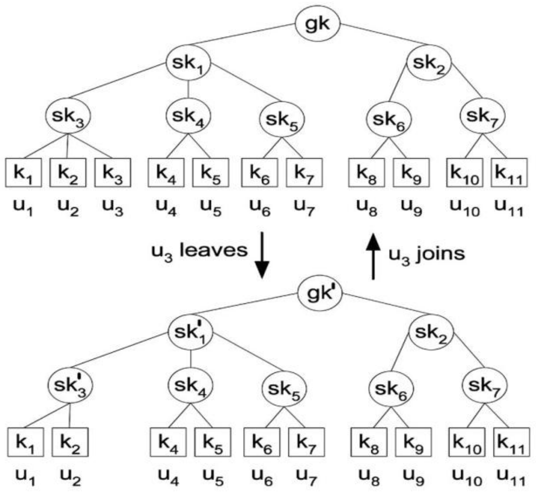
gk
sk1
sk2
sk3
sk4
sk5
sk6
sk7
k1 k2 k3 k4 k5 k6 k7 k8 k9 k10 k11
u1 u2 u3 u4 u5 u6 u7 u8 u9 u10 u11
u3 leaves
u3 joins
gk'
sk1'
sk3'
k1 k2 k4 k5 k6 k7 k8 k9 k10 k11
u1 u2 u4 u5 u6 u7 u8 u9 u10 u11

Fig.1

Group oriented multi-cast can be achieved successfully by using shared group key. Eskicioglu has proposed an overview of the schemes for group key management, authentication and watermarking in wired networks with fixed members and wireless networks with mobile members [3]. Based on a novel application of one-way function trees, a new scalable centralized algorithm, called OFT was presented by Sherman and McGrew, in [15], for establishing shared cryptographic keys in large, dynamically changing groups. Wong et al address the scalability problem of multicast key management [19]. Rafaeli and Hutchison [13] has presented a survey of group key management and classified the proposed solutions into three main classes: centralized group key management protocols, decentralized architectures and distributed key management protocols.

To alleviate the problems of rekeying after a single user join/leave, batch rekeying was proposed. The use of periodic batch rekeying which can improve efficiency and alleviate the out-of-sync problem was discussed by Li et al [7].

2.2 Balanced key tree

The number of keys stored by a user in the key tree and its efficiency depend on whether the key tree is balanced. In an unbalanced binary key tree, the number of keys that the users store varies from three to six for a group of eight users. Of these, u1 and u2 store six keys, which in turns post an overhead of five decryptions when a user departs. If the key tree is balanced the same set of eight users need just four keys and three decryptions in case of leave. Therefore a balanced key tree benefits the storage and the rekeying cost.

Josep Pegueroles [6] has presented a technique for multicast batch rekeying, which reallocates the tree nodes in order to keep the tree balanced all the time. In [11], Ng and Zhili Sun discussed the efficiency of Logical Key Hierarchy which depends on whether the key tree remains balanced. Scalable group re-keying for secure multicast, based upon the idea of periodic group re-keying was discussed by Sanjeev Setia et al [14]. Two Merging Algorithms suitable for batch join events were proposed by Wee Hock Desmond Ng et al in [18]. The design and specification of a protocol based upon the use of key trees for secure groups and periodic batch rekeying, for scalable and reliable group rekeying together with performance evaluation results was provided by Zhang et al in [21].

Zhang and Wang [22] have introduced a centralized key management scheme for hierarchical access control that considers both partially ordered users and partially ordered data streams and that improves the efficiency of key management by encrypting multiple equivalent data streams with a single data encryption key, instead of encrypting each data stream with a unique data encryption key in the multi-group key management Scheme.

2.3 AVL trees approach

Rodeh *et al* in [12] proposes a fault tolerant group key management by using AVL trees for managing group keys in group communications system. It uses collaboration of users to create the group key graph unlike centralized approach. But the approach fails to support forward secrecy only in a strong sense and also has a scalability problem when the group size increases.

2.4 Height balanced 2-3 tree and its approach

A height balanced 2-3 trees are special case of B-trees where, all leaves are at the same depth and all internal nodes have out degree two or three.

The nodes that represent group members are external nodes, shown as square node in figure 4 and all other nodes are internal nodes.
The level of a node x is defined as $x.level = x.parent.level + 1$ and $root.level = 0$

Goshi and Ladner in [5] have analysed height and weight 2-3 trees and concluded that height-balanced 2-3 had the best performance. The number of multicasts is increased when there is node splitting with a user join in a Height balanced 2-3 tree.

Lu, H [8] has introduced a special class of trees called NSBHO trees as shown in **Figure 3**. It does not need node splitting during a user join.

2.5 Non-Split Balancing Higher Order (NSBHO) tree

An empty tree is an NSBHO tree of order m. A tree with only one external node and no internal nodes is an NSBHO tree of order m. If an NSBHO tree of order m is not empty and has more than one external node, it has the following properties:
P1. The root has at least two children and at most m children.
P2. All external nodes are at the same level.
P3. All internal nodes other than the nodes in the special path (defined below) and the root have at least $d = \lceil m/2 \rceil$ children and at most m children.
P4. There is at most one special path.
P5. A special path (SP) is a sequence of internal nodes, $(z_0, z_1, .., z_k)$, where $z_i$is an ancestor of z_{i+1} for $0 \leq i < k, z_i$ has at least one child and at most d-1 children for $0 \leq i \leq k$, and $z_0$is not the root.
A sample NSBHO tree of order 3 is shown in fig 2.

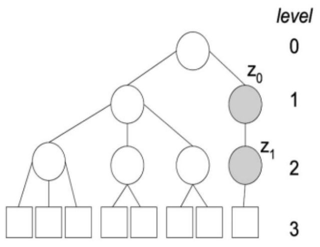

Fig.2 NSBHO tree of order 3

2.6 Key Management in Multi- Privileged Groups

In conventional group communication scheme, all members in a group have same level of access privileges. But many group applications have multiple related data streams and members have different access privileges as shown in Fig. 3. Multi- Group key management scheme was proposed for such a case. There are two challenging factors involved in this scheme. Users can subscribe to one or more data streams which are encrypted by separate Session Keys. The challenge inherent here is to ensure that the users cannot access beyond their privileges. The second factor is to provide a flexible group key management scheme when the users join/change /leave their requirements.

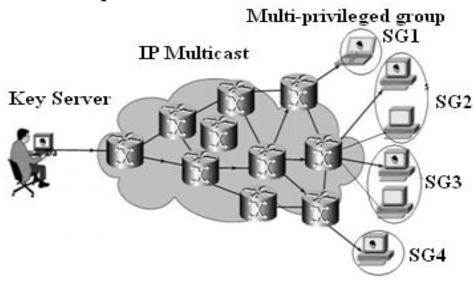

Fig. 3: Internet Protocol Multicast in Multi-privileged group

When the users have different access privileges with the usage of multiple resources, multi-privileged groups came into existence. An integrated key graph that maintains keying material for all members with different access privileges in multi-group key management scheme that achieves hierarchical group access control was given by Sun and Liu [14]. A dynamic access control scheme for group communications which support multiple service groups with different access privileges was proposed by Ma et al in [7]. A centralized key management scheme for hierarchical access control that considers both partially ordered users and partially ordered data streams, which improves the efficiency of key management by encrypting multiple equivalent data streams with a single data encryption key, instead of encrypting each data stream with a unique data encryption key in the multi-group key management scheme was proposed by Zhang and Wang in [20]. An efficient group key management scheme called ID-based Hierarchical Key Graph Scheme (IDHKGS) for secure multi-privileged group communications which employs a key graph, on which each node is assigned a unique ID according to access relations between nodes, was proposed by Wang et al [15].

In various scenarios Multi-privileged group key management plays a very important role. It provides a service by which a television audience can purchase events to view via private telecast of that event to their homes. The broadcaster shows the event at the same time to everyone ordering it. Events often include feature films, sporting events and special events. In military group communications, which is hierarchically managed, participants have different access authorizations [20]. In e-newspaper broadcasting, there are multiple data streams to broadcast the contents of top news, weather forecasts, financial news, stock quotes and sports news. The service provider also classifies users into several membership groups, such as gold, silver sports, silver finance and basic. In this application members in different groups can access different contents [22].

Let $\{r_1, r_2, r_3 \ldots.\}$ denote the set of resources in the group communication system. A Data Group (DG) consists of the users who can access a particular resource and a Service Group (SG) consists of users who are authorized to access exactly the same set of resources. The DGs have overlapped membership while the SGs don't. The DGs are denoted by $D_1, D_2, D_3 \ldots. D_M$ and SGs are denoted by $S_1, S_2, S_3 \ldots. S_I$ where M and I are the total number of DGs and SGs respectively.

Example:

Consider the following scenario where the resources are, News (r_1), Stock quote (r_2) and Traffic/Weather (r_3). Then the users can subscribe any combination of the resources which are the Service Groups (SGs):. Thus, there are a total of seven SGs and three DGs, denoted by $S_1, S_2, S_3 \ldots. S_7$ and D_1, D_2, D_3 respectively as listed in table 1

Table 1

Access Relation	$D_1 access\{r_1\}$	$D_2 access\{r_2\}$	$D_3\ access\{r_3\}$
$s_{\{001\}}$	✓		
$s_{\{010\}}$		✓	
$s_{\{011\}}$	✓	✓	
$s_{\{100\}}$			✓
$s_{\{101\}}$	✓		✓
$s_{\{110\}}$		✓	✓
$s_{\{111\}}$	✓	✓	✓

In [3] Wang et al proposed an ID- based hierarchical key graph scheme for the multi-privileged users [3]. It contains two types of nodes; u-node to have individual keys and k-nodes to have SG keys, DG keys and auxiliary keys. If a user knows the u-node ID in the group then he can deduce the IDs of k-nodes on the path from u-node to SK (Session Key) nodes.The key graph contains two parts namely, SG part and DG part where the former has all SG- sub trees and the later has all SK-nodes and k-nodes between SG nodes and SK nodes, as shown in fig. 4. S2, S3….Si denotes the SGs; where i is a prime number. A node in the SG subtree is identified by <i,m> where 'i' denotes as to which SG subtree the node belongs and m denotes the position of that node in that SG subtree. <i,0> denotes the root of the SG subtree Si. The nodes are numbered from roots of the SG subtree in top-down, left-right order.

In DG part, if a node has

i) Two children, $<j_1, n_1>$ and $<j_2, n_2>$, then this node is identified by <j,n>,where $j=lcm(j_1, j_2)$ and $n=max(j_1, j_2)$.
ii) Exactly one child, $<j_1, n_1>$, then it is identified by $<j_1, -1>$

When the users join the binary key tree, rekeying is done in order to preserve forward secrecy. Each rekey message has to be signed to preserve authentication and signing is computational expensive.

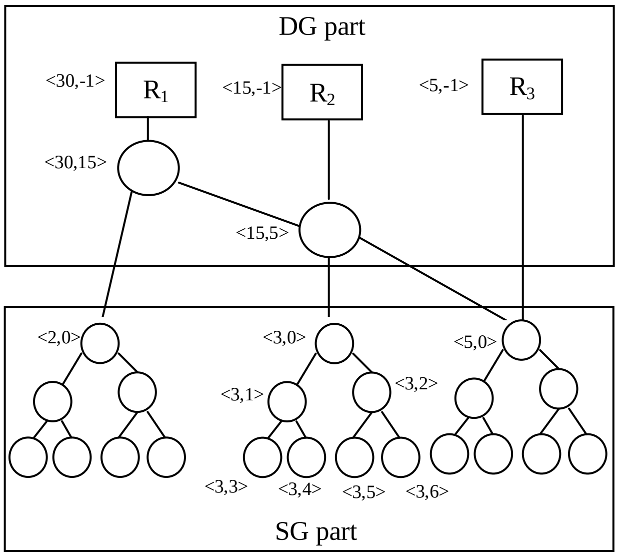

Fig. 4 A key graph representation using IDHKGS

The number of nodes key encryption keys that are getting affected when a new user joins the key tree increases with the size of the key tree. Also it poses a threat on the number of multi-cast messages that need to be sent to the existing users. B-trees were used so that the number of users that can be accommodated under a key encryption key is more than two to alleviate this drawback of the binary trees. But even with B-trees the problem still pertains since there is node splitting during user join.

```
Procedure User_join_height_balanced ( Usernode u, root id of SG tree )
  Select a node p that causes minimum increase in the tree weight when used as insertion point, to insert the new member.
If p has two children then
        Add u as child of p and stop procedure
endif
Create a new node p'
Borrow a child from p
Insert the borrowed child and u as the children of p'
If p has no parent then
        create a new root r ;
        Insert p and p' as its children;
    else
        call procedure user_join_height_balanced recursively to add p' as a child of the parent g;
endif
End User_join_height_balanced
```

Algorithm 1: For user join in multi-privileged key tree using height balanced 2-3 tree

```
Procedure User_leave_height_balanced()
If p, the parent of u has three children then
        remove u
 else
        if p is root, then
                remove p and u, leaving the sibling of u as the new root;
        else
                remove u
        endif
endif
    If s, the sibling of p has three children then
        grant one child to p
    else
        add the sibling of u as the child of s and repeat procedure recursively to delete p
    endif
End User_leave_height_balanced
```

Algorithm 2. For user leave in multi-privileged key tree using height balanced 2-3 tree

Fig. 5 shows the user join/leave operations in a height balanced 2-3 tree. The users U6,U7 and U8 share the key K as shown in Fig.5(a). When the user U9 joins the group, this key node is split and a parent node(Shaded black) is created as shown in fig.5(b). The key K in fig.5(a) cannot be used any further and the user pairs (U6,U7) and (U8,U9) share a different set of newly created keys, L and M respectively . All these four users share the key of the new parent shown in black in Fig.5(b). All the other parent nodes of U9 are also affected due to this join.

When U10 joins its immediate parent L is changed to N, but not split and all the other parent nodes are getting affected.

When U9 joins the number of unicasts is more as compared to the case of U10 join. In the former case there is node splitting. This results in an additional unicasting to inform the keys L and M to users (U6,U7) and (U8,U9) respectively.

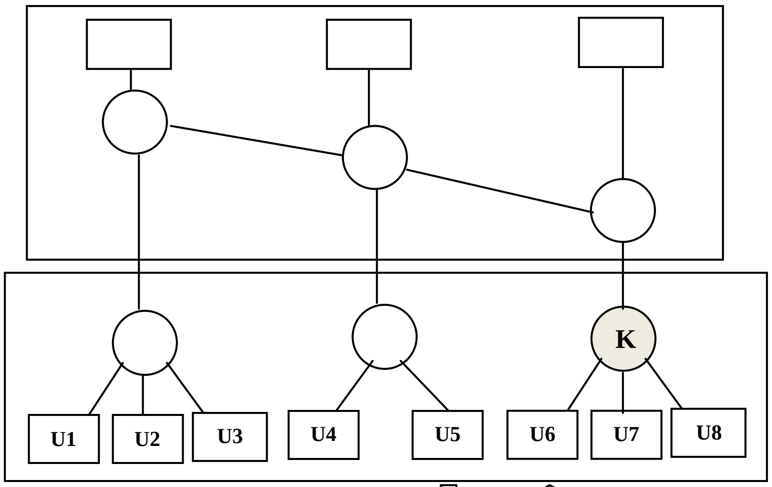

Fig. 5 (a) U9 Joins ⇩ ⇧U9 Leaves

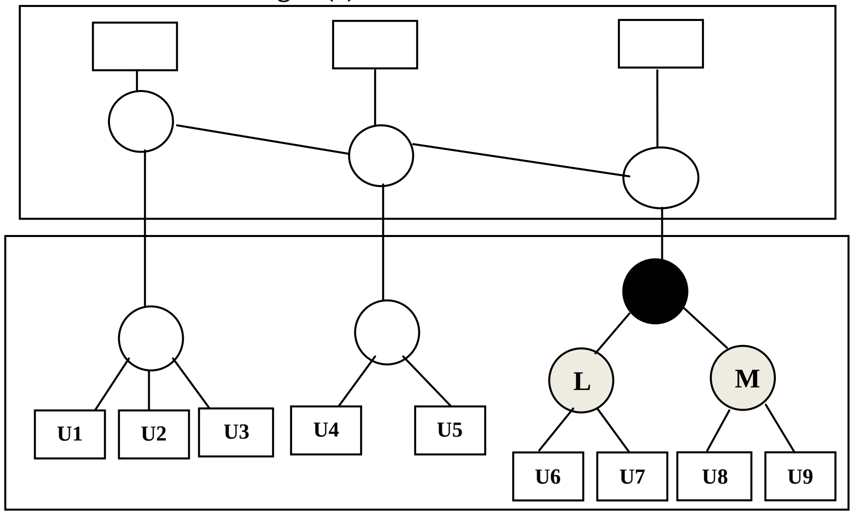

Fig. 5 (b) U10 Joins ⇩ ⇧U10 Leaves

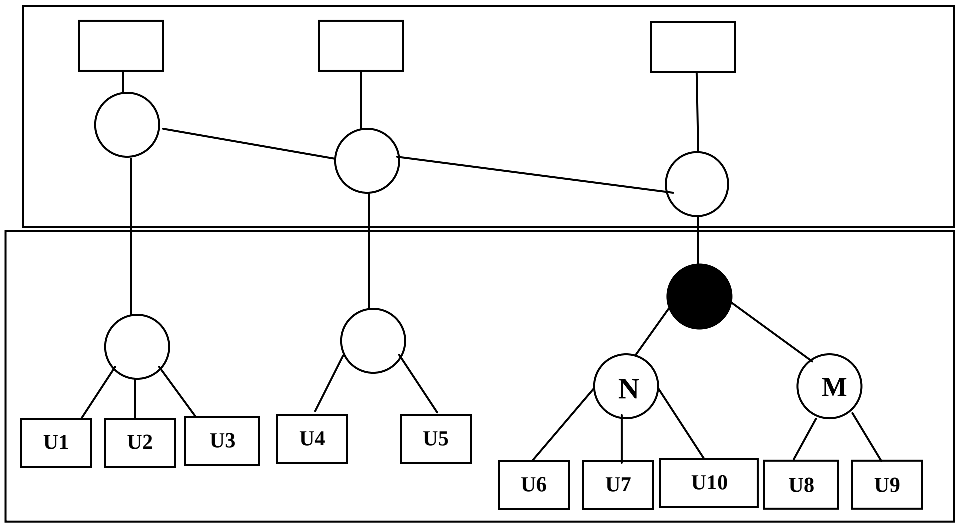

Fig. 5 (c)

Fig.5 A key graph representation using height balanced 2-3 tree

```
Procedure Join_ NSBHO(node z)
        If root is NULL then
                set root as z;
                 stop procedure;
        else
                        If no internal node is found or all internal nodes are full then
                        Create a new node y and set the current root as child of y and set root as y
                        else
                           If SP is not empty then
                                   set y as the node in SP with largest level
                           else
                                   set y as a non-full internal node that is not in SP
                           endif
                        endif
         endif
        If level of y is h-1 then
                    Set z as child of y
        Else
                        If SP is full or SP is not found Create a chain of internal nodes $(x_0, x_1, \dots x_l)$ where $x_i$
                        is parent of $x_{i+1}$ and $0 \leq i < l$
                        Level of $x_0$ is (level of y)+1
                        Level of $x_l$ is h-1
                         Set z as child of $x_l$ and $x_0$ as child of y
       endif
 end Join_NSBHO
```

Algorithm 3. For Join in multi-privileged key tree using NSBHO

```
//        If z is in SP then one child short means it does not have any child else it has either one child or no
children at all.
Procedure remove_NSBHO(Z)
If z is root then
    Set root as null
    Stop Procedure_remove_NSBHO
endif
Set parent of z as pz
Remove the child z from pz
 set pz as z
while z is not root and z is one child short then
                set parent of z as pz
                If z belongs to SP then
                Remove z from SP
                delete it and set pz as z
                Else
                        If a sibling of z belongs to SP and has size>1 then
                                set it as sz
                        Else
                                If  a sibling of z that is not in SP has size > d then
                                        set it as sz
                                Else
                                        Set null as sz
                                Endif
                        Endif

                        If sz is not null then
                                move a child from sz to z
                                 If z is root and size of z is less than two then
                                        Set the only child of z as root
```

```
                    if root belongs to sp then remove root from sp
                Endif
                If size of pz is greater than one then
                    merge z with a sibling of z and set pz as z
                else
                    z is added to sp and set root as z
                endif
        endif
end while
end remove_NSBHO
```

Algorithm 4. For removing Z from multi-privileged key tree using NSBHO

Algorithm 5 and 6 respectively are the rekeying algorithms for join and leave cases.

```
Procedure Rekey_Join()
Mark the parents of $< i, m >_J$ till the root $< i, 0 >$
If the key node is not newly created then
        $K' < i, j > = f(K < i, j >)$
If the key node is root, then
        assign a new key $K' < i, 0 >$

else
        compute $K' < i, j > = f(K < i, s > \oplus K < i, s >)$
    Where $< i, s >$ is the left sibling of $< i, j >$
endif
Set n as i of $< i, m >_J$
Collect those nodes with non-prime i from DG part and also the node that has i same as n
If i mod n is zero then compute $K' < i, j > = f(K < i, j >)$
End Rekey_Join
```

Algorithm 5. Rekeying Algorithm for join

```
Procedure Rekey_Leave()
Mark the affected nodes in the leave path
If the affected node has leaf nodes then the new key is computed using
        $K' < i, j > = f(K < i, j > \oplus K_L < i, j >)$
        where $K_L < i, j >$ denotes the key of the left most child of $< i, j >$
    Else
        $K' < i, j > = f(K < i, j > \oplus K_{UL} < i, j >)$
        where $K_{UL} < i, j >$ denotes the key of the left most unaffected child of $< i, j >$
endif
Collect those nodes with non-prime i from DG part and also the node that has i same as n;
If i mod n is zero then
    compute $K' < i, j > = f(K < i, j > \oplus K_c < i, j >)$
    Where $K_c < i, j >$denotes the key of the child (unaffected if present;  the only child otherwise) of $< i, j >$
End of Procedure_Rekey_Leave
```

Algorithm 6. Rekeying Algorithm for leave

Consider the case when users U11 to U14 join in fig.5(c). It does not need any node splitting. When user U15 joins, the node marked black in fig.5(c) will be split into two nodes, resulting in multicasting those keys to the users U6 to U14. Hence whenever there is node splitting the number of unicasts/multicasts increases depending on the level in which the node is split.
NSBHO approach does not need node splitting and hence reduces the number of multicasts as shown in fig.6.When U9 joins fig.6(a), the key node K is not disturbed at all and a special path shown in grey is created (Fig.6(b)). Also when U10 joins, it joins the same special path without affecting K as shown in Fig. 6(c).

Hence as compared to 2-3 trees, the usage of NSBHO trees reduces the number of multicasts/ unicasts due to node splitting.

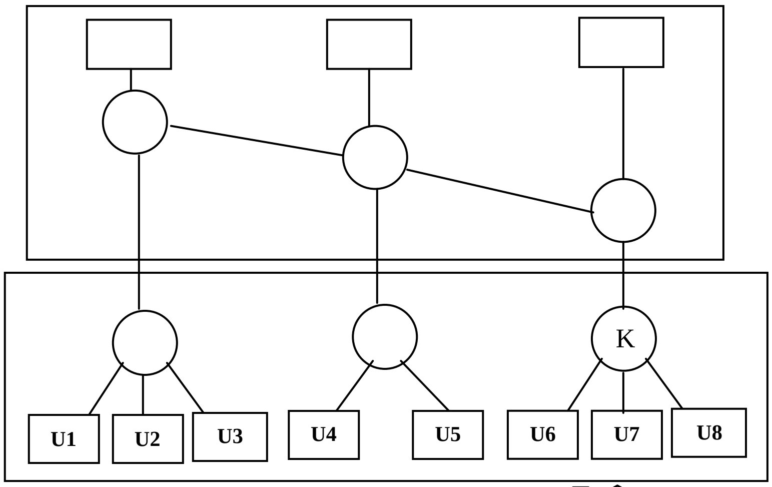

Fig. 6 (a) U9 Joins ⇩⇧ U9 Leaves

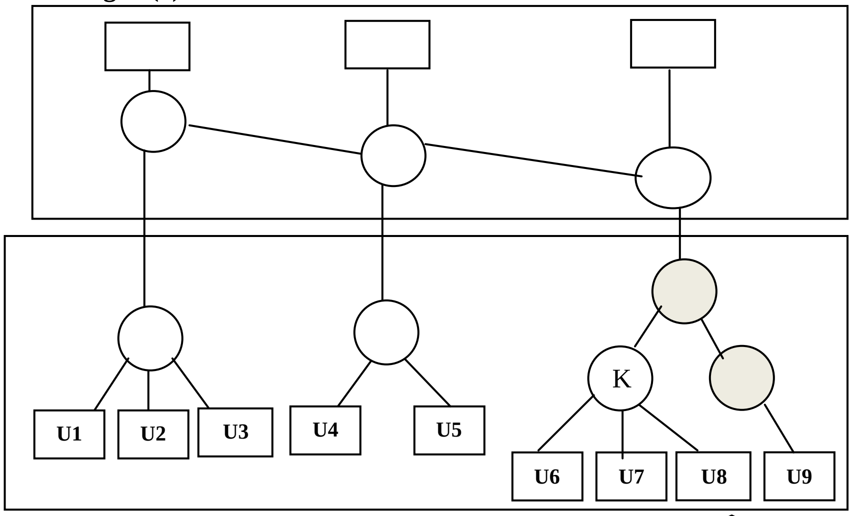

Fig. 6 (b) U10 Joins ⇩ ⇧U10 Leaves

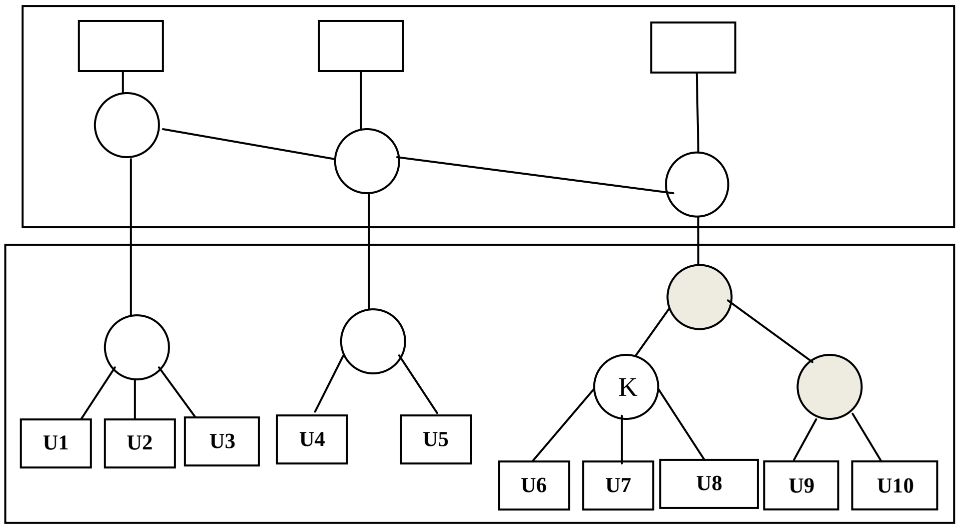

Fig. 6 (c)

Fig.6. A key graph representation using NSBHO tree

4.Discussion

This section gives the analysis on the bounds for the number of external nodes in key graph representation using Binary trees, height balanced 2-3 trees and NSBHO trees. It also discusses about the rekeying message costs of Binary tree, B-Tree and the NSBHO Tree during user join and leave.

Lemma1: (Binary Tree)

Let h_{max} and h_{min}be the maximum and minimum heights of the SG part and n be the number of external nodes then the bounds for n is given by

$$\left(2^{h_{min}-1}+1\right)l \leq n \leq \left(2^{h_{max}}\right)l$$

Proof:

To prove: $\boldsymbol{n \leq \left(2^{h_{max}}\right)l}$

When there are no users then the sub-trees are at level zero and there are l nodes.When all the users are at the first level, there are at most $2l$ nodes, since each node in level zero can have at most two children. Similarly when all the users are at level i-1, there are at most $l2^{i-1}$ children. At level i, there are at most $(lm^{i-1}) * m = lm^{i}$ children, since each node of the previous level can have at most two children at this level. Therefore, level h_{max} has at most $l2^{h_{max}}$ children

To prove: $\boldsymbol{\left(2^{h_{min}-1}+1\right)l \leq n}$

When there are no users then the sub-trees are at level zero and there are at least l nodes. When all the users are at the first level, there are at least $(2^{1-1}+1)l = 2l$, users since each node in level zero should have at least two children. Level $i-$ 1must have atleast $\left(2^{i-2}+1\right)l$. Level i must have atleast $2^{i-2}l * 2 + l = \left(2^{i-1}+1\right)l$ users.

Lemma2:

$$\boldsymbol{log_2 n \leq h \leq log_2(n-1)+1}$$

Proof: The proof is direct from the previous lemma.

Lemma3: (2-3 tree)

Let h_{max} and h_{min}be the maximum and minimum heights of the SG part and n be the number of external nodes then the bounds for n is given by

$$2ld^{h_{min}-1} \leq n \leq lm^{h_{max}}$$

where l denotes the number of SG trees

$d = \lceil m/2 \rceil$

Proof:

The result is proved using mathematical induction.

To prove : $\boldsymbol{n \leq lm^{h_{max}}}$

When L= 0, there are l nodes (The roots of SG trees)

When L= 1, there are at most lm nodes, since each node in level zero can have at most m children

Assume when L= i-1, there are at most lm^{i-1} children

To prove L= i has at most lm^{i} children

At L=i, there are at most L(i-1)*m children, since each node of the previous level can have at most m children at this level.

Hence, L(i)=L(i-1)*m= $(lm^{i-1}) * m = lm^{i}$

Therefore, level h_{max} has at most $lm^{h_{max}}$ children

To prove: $\boldsymbol{2ld^{h_{min}-1} \leq n}$

When L= 0, there are l nodes (The roots of SG trees)

When L= 1, there are at least $2l$ nodes, since each node in level zero should have at least 2 children

When L= 2, there are at least $2ld$ nodes, since each node in level 1 should have at least d children

Assume when L= i-1, there are at least $2ld^{i-1}$ children

To prove L= i has at least $2ld^{i}$ children

At L=i, there are at least L(i-1)*d children, since each node of the previous level should have at least d children at this level.

Hence, L(i)=L(i-1)*d= $(2ld^{i-1}) * d = 2ld^{i}$

Therefore, level h_{min} has at least $2ld^{h_{min}}$children

Lemma 4:

$$log_d(n/2l) + 1 \geq h \geq log_m(n/l)$$

Proof: The proof of this is obvious from the previous lemma.

Lemma5 :(NSBHO)

Let h_{max} and h_{min}be the maximum and minimum heights of the SG part and n be the number of external nodes then the bounds for n is given by

$$l(d^{h_{min}-1}+1) \leq n \leq lm^{h_{max}}$$

where l denotes the number of SG trees

$$d = \lceil m/2 \rceil$$

$h_{min} = \min(h_j)$;$h_{max} = \max(h_j)$ where h_j is the height of $SG_j$where j=2,3,…

Proof:

To prove $n \leq lm^{h_{max}}$

When there are no users then the sub-trees are at level zero and there are l nodes.When all the users are at the first level, there are at most lm nodes, since each node in level zero can have at most m children. Similarly when all the users are at level i-1, there are at most lm^{i-1} children. At level i, there are at most $(lm^{i-1}) * m = lm^i$ children, since each node of the previous level can have at most m children at this level. Therefore, level h_{max} has at most $lm^{h_{max}}$ children

To prove $l(d^{h_{min}-1}+1) \leq n$

When there are no users then the sub-trees are at level zero and there are at least l nodes. When all the users are at the first level, there are at least $2l$ nodes, since each node in level zero should have at least two children (as per definition of NSBHO). At level 1 there are $2l$ nodes of which at least l are on the Special Path and a special path can have at least one child as per definition and hence these l nodes will have at least $l$children at level 2. The remaining $l$nodes that are not in Special path can have at least d children and hence there will be at least dl children at level 2 for these l nodes. Hence total number of external nodes at level 2 is at least $l + dl$. Similarly when all the users are at level i-1, the number of external nodes is at least $l(d^{i-2}+1)$. At level i, the number of external nodes in the special path is at least l and for the nodes that are not in the special path is at least$(ld^{i-2})d$. Hence the total no. of external nodes at level i is at least $l(d^{i-1}+1)$.When L= 0 there are at least l children.

Lemma 6:

$$log_d\left(\frac{n}{l}-1\right)+1 \geq h \geq log_m\left(\frac{n}{l}\right)$$

Proof: The proof of this is direct from the previous lemma.

RESULT:

Difference between the worst case heights of B-tree and NSBHO trees of order m, for Multi-Privileged groups is

$(log_d\left(\frac{n}{l}-1\right)+1) - (log_d(n/2l)+1) = log_d(2(n-l)/n)$

When $n \gg 1$, $log_d(2(n-l)/n) = log_d 2 \leq 1$. Hence SG part of an NSDHO is at most one level taller than the SG part of B-tree of the same order in Multi-Privileged groups.

Table.2 gives the rekeying message costs for insertion of a node in a Binary tree, B-tree and NSBHO tree for Multi-privileged groups.

Table.2.Rekeying message costs for insertion

Insert	Best	Worst
Binary Tree	2(h+1)	$3l + 2(h-1)$
2-3 Tree	2(h+1)	$2(h+l) + mh - 1$
NSBHO	h+3	$2(h+l) - 1$

The best case rekeying cost for all the three cases happens when the user joins the first subgroup, where two nodes in the DG part are affected.

When Binary trees are used, if h is the height of the service group tree then multicast of size h need to made and a unicast of size h+2 is made. Thus the best case rekeying cost comes to 2h+2. In the worst case, the size of unicasts is $(2l-1)+h$ and the size of multicast is $h+l-1$ and hence the worst case cost is $3l+2(h-1)$.

Using 2-3 trees, if h is the service-group tree height then the multicast size is h and one unicast of size h+2 is made to the newly joined user. The worst case rekeying cost happens when there is node splitting due to a user join and the user joins at the last service group. In this case the number of nodes affected in the SG part is mh+2h and the number of nodes affected in the DG part is 2l-1, putting the worst case rekeying cost to$2(h+l)+mh-1$.

Similarly the best case rekeying cost in NSBHO tree is h+3 where the user joins the first service group and a new root is created. In this case there is only one multicast to inform the new root, using the existing root of the subtree and one unicast to inform the new user the existing keys. When the user joins the first service group two nodes in the DG part gets affected. The worst case rekeying cost occurs in the case when the user joins the last service group at an insertion point h-1 where h is the height of the last service group. The rekeying cost for this case is $2(h+l)-1$.

Table.3.Rekeying message costs for deletion

Delete	Best	Worst
Binary Tree	h	$h+2l-3$
2-3 tree	$(h-1)d+4$	$d+mh+2(l-1)$
NSBHO	3	$d+mh+2(l-1)$

The rekeying costs for leave case are also listed in table.3 and the case where user leaves from the first service group results in best case rekeying cost and the user leave from the last service group results in the worst case rekeying cost, when the service groups are considered in left right order.

5. Conclusion and future work

Key management is important in group communications and this paper has dealt with managing keys using B-trees and NSBHO trees for multi-privileged groups. Algorithms for join and leave in both B-trees and NSBHO trees are provided and rekeying algorithms for the same respectively have also been developed. Bounds for the tree heights have been proved as lemmas. A performance comparison of the Multi-privileged key management using B-trees and NSBHO trees has been provided which shows that NSBHO gives better results. The applications discussed can be implemented as future work using NSBHO trees.

References

[1] Banerjee, S.; Bhattacharjee, B., Scalable secure group communication over IP multicast, *IEEE Journal on Selected Areas in Communications*, vol.20, no.8, pp. 1511-1527, Oct 2002.

[2] D.Ma,Y.Wu,R.Deng,T.Li, Dynamic access control for multi-privileged group communications,Proceedings of 6th International Conference on Information and Communications Security, *Lecture Notes in Computer Science(LNCS)*,3269(2004)508-519.

[3] Eskicioglu. A.M., Multimedia Security in Group Communication: Recent Progress in Key Management, Authentication and Watermarking, *ACM Multimedia Systems J., special issues on multimedia security*, pp. 239-248, Sept. 2003.

[4] Guojun Wang; Jie Ouyang; Hsiao-Hwa Chen; Minyi Guo, ID-Based Hierarchical Key Graph Scheme in Multi-Privileged Group Communications, *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE* , vol., no., pp.172-176, 26-30 Nov. 2007.

[5] Justin Goshi and Richard E. Ladner. 2003. Algorithms for dynamic multicast key distribution trees. In *Proceedings of the twenty-second annual symposium on Principles of distributed computing* (PODC '03). ACM, New York, NY, USA, 243-251.

[6] Josep Pegueroles, Francisco Rico-Novella, Balanced Batch LKH: New Proposal, Implementation and Performance Evaluation., iscc, *Eighth IEEE Symposium on Computers and Communications, 2003*. vol.2 pp. 815 - 820 vol.2

[7] Li, X. S., Yang, Y. R., Gouda, M. G., and Lam, S. S. 2001. Batch rekeying for secure group communications. *In Proceedings of the 10th international Conference on World Wide Web* (Hong Kong, Hong Kong, May 01 - 05, 2001). WWW '01. ACM, New York, NY, 525-534.

[8] Lu, H. 2005. A Novel High-Order Tree for Secure Multicast Key Management. *IEEE Trans. Comput. 54*, 2 (Feb. 2005), 214-224.

[9] Moyer.M.J., Rao J.R., and Rohatgi .P., "Maintaining Balanced Key Trees for Secure Multicast," *Internet Research Task Force (IRTF), Internet draft, draft-irtf-smug-key-tree-balance-00.txt*, June 1999.

[10] Ng, W. H., Howarth, M., Sun, Z., and Cruickshank, H. 2007. Dynamic Balanced Key Tree Management for Secure Multicast Communications. *IEEE Trans. Comput.* 56, 5 (May. 2007), 590-605.

[11] Ng, W.H.D.; Zhili Sun, "Multi-layers balanced LKH," *Communications, 2005. ICC 2005. 2005 IEEE International Conference on* , vol.2, no., pp. 1015-1019 Vol. 2, 16-20 May 2005.

[12] O. Rodeh, K. P. Birman, and D. Dolev. Using AVL trees for fault tolerant group keymanagement. *International Journal on Information Security*, pp 84-99, 2001.

[13] Rafaeli, S. and Hutchison, D. 2003. A survey of key management for secure group communication. *ACM Comput. Surv*. 35, 3 (Sep. 2003), 309-329.

[14] Sanjeev Setia , Samir Koussih , Sushil Jajodia , Eric Harder, Kronos: A Scalable Group Re-Keying Approach for Secure Multicast, *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, p.215, May 14-17, 2000

[15] Sherman, A. T. and McGrew, D. A. 2003. Key Establishment in Large Dynamic Groups Using One-Way Function Trees. *IEEE Trans. Softw. Eng*. 29, 5 (May. 2003), 444-458.

[16] Sun, Y. and Liu, K. J. 2007. Hierarchical group access control for secure multicast communications. *IEEE/ACM Trans. Netw.* 15, 6 (Dec. 2007), 1514-1526.

[17] Wang, G., Ouyang, J., Chen, H., and Guo, M. 2007. Efficient group key management for multi-privileged groups. *Comput. Commun* .vol.30, 11-12 (Sep. 2007), pp 2497-2509.

[18] Wee Hock Desmond Ng, Haitham S. Cruickshank, Zhili Sun:, "Scalable Balanced Batch Rekeying for Secure Group Communication," *Elsevier Computers and Security*, vol. 25, pp. 265-273, June 2006.

[19] Wong, C. K., Gouda, M., and Lam, S. S. 2000. Secure group communications using key graphs. *IEEE/ACM Trans. Netw.* 8, 1 (Feb. 2000), 16-30.

[20] Yan Sun; Liu, K.J.R., "Scalable hierarchical access control in secure group communications," *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies* , vol.2, no., pp. 1296-1306 vol.2, 7-11 March 2004.

[21] Zhang, X. B., Lam, S. S., Lee, D., and Yang, Y. R. 2003. Protocol design for scalable and reliable group rekeying. *IEEE/ACM Trans. Netw.*11,6(Dec.2003),908-922.

[22] Zhang.Q.,Wang.Y., A centralized key management scheme for hierarchical access control, *Proceedings of Global Telecommunications Conference4 (2004)* 2067-2071.

Ms.A. Muthulakshmi has six years of teaching experience and has guided five PG projects. She has attended 10 National/International conferences and workshops in the area of cryptography and security in computing. She is a member of Cryptology Research Society of India.

Dr. R. Anitha is an Associate Professor in the Department of Mathematics and Computer Applications. She has 24 years of teaching experience. She has guided 3 PhDs and 1 M.Phil in Applied Mathematics. At present she is guiding 9 research scholars. She is the Program Coordinator of the five year integrated M.Sc. Theoretical Computer Science programme. Currently she is the principal investigator (PSG Tech) of the CDBR-Smart and Secure Environment project, funded by NTRO, which is a collaborative research work of eight Institutions. She has rendered seven years of service for NSS. She is a member of ISTE, CRSI and ACM. She has visited France and Australia on academic grounds .Cryptography and Security in Computing are her areas of interest.

Ms. M. Sumathi is a Lecturer in the Department of Mathematics and Computer Applications. She has three years of teaching experience. Her area of interest is Optimization Techniques.