Distributed Group Key Management with Cluster based Communication for Dynamic Peer Groups

Rajender Dharavath[1]
Department of Computer Science & Engineering
Aditya Engineering College
Kakinada, India.
rajendar.dharavath@gmail.com

K Bhima[2]
Department of Computer Science & Engineering
Brilliant Institute of Engineering &Technology
Hyderabad, India.
bhima_mnnit@yahoo.co.in

***Abstract*—Secure group communication is an increasingly popular research area having received much attention in recent years. Group key management is a fundamental building block for secure group communication systems. This paper introduces a new family of protocols addressing cluster based communication, and distributed group key agreement for secure group communication in dynamic peer groups. In this scheme, group members can be divided into sub groups called clusters. We propose three cluster based communication protocols with tree-based group key management. The protocols (1) provides the communication within the cluster by generating common group key within the cluster, (2) provides communication between the clusters by generating common group key between the clusters and (3) provides the communication among all clusters by generating common group key among the all clusters. In our approach group key will be updated for each session or when a user joins or leaves the cluster. More over we use Certificate Authority which guarantees key authentication, and protects our protocol from all types of attacks.**

Keywords- Secure Group Communication; Key Agreement; Key Tree; Dynamic peer groups,Cluster.

I. INTRODUCTION

As a result of the increased the popularity of group oriented applications such as pay-TV, distributed interactive games, video and teleconference and chat rooms. There is a growing demand for the security services to achieve the secure group communication. A common method is to encrypt messages with a group key, so that entities outside the group cannot decode them. A satisfactory group communication system would possess the properties of group key security, forward secrecy, backward secrecy, and key independence [1,2,3].

In this paper research efforts have been put into the design of a group key management and three different cluster based communication protocols. There are three approaches for generating such group keys: centralized, decentralized, and distributed. Centralized key distribution uses a dedicated key server, resulting in simpler protocols. However, centralized methods fail entirely once the server is compromised, so that the central key server makes a tempting target for adversaries. In addition, centralized key distribution is not suitable for dynamic peer groups, in which all nodes play the same function and role, thus it is unreasonable to make one the key server, placing all trust in it. In the decentralized approach, multiple entities are responsible for managing the group as opposed to a single entity. In contrast to both approaches, the distributed key management requires each member to contribute a share to generate the group key, resulting in more complex protocols. And each member is equally responsible for generating and maintaining the group key.

In this paper the group key or common key is generated based on distributed key management approach. The group key is updated on every membership change, and for every session, for forward and backward secrecy [1, 2, 3], a method called group rekeying.

To reduce the number of rekeying operations, Woung.et al [7] proposed a logical data structure called a key tree. And Kim et al [1], proposed a tree-based key agreement protocol, TGDH which is combination of key tree and Diffie-Hellman key to generate and maintain the group key. But it suffers from the impersonation attack because of not regularly updation of keys and generates unnecessary messages. Based on above two ideas Zhou, L., C.V. Ravishanker and Kim et al [6], proposed an AFTD (Authenticated Fault-tolerant Tree-based Diffie-Hellman key exchange Protocol) protocol, which is the combination of key trees and Diffie-Hellman key exchange for group key generation.

Assume that the total network topology considered as a group, which can be divided into subgroups called clusters. Group is divided into clusters based on the location identification number; LID's of users, and cluster is assigned with cluster identification numbers, CID, which are given by the Certificate Authority, CA at the time of user joining into cluster or group. Issuing location identification number and public key certificate to the new user are the offline actions performed by the certificate authority, CA.

Each cluster member maintains its own cluster key tree and generates the cluster group key for secure communication. We assume in every cluster, every node can receive a message broadcasted from the other nodes. Each cluster is headed by a cluster head or sponsor of cluster and he is responsible for generating cluster group key, who is shallowest rightmost to the user (in cluster key tree) joins or leaves from the cluster.

Cluster group key or cluster common key is shared by all the cluster members and communicates with it. The authentication is provided by certificate authority by issuing the

public key certificate and location identification number, LID prior to the time of joining in the cluster or group.

The rest of the paper is organized as follows. Section 2 focuses on related work in this field. We present our proposed scheme in Section 3, communication protocols and group key management techniques are discussed in Section 4. Dynamic network peer groups are presented in Section 5, security analysis in section6. Finally we make a conclusion in Section 7.

II. RELATED WORK

Key trees [6] were first proposed for centralized key distribution, while Kim et al.[1], adapted it to distributed key agreement protocol TGDH. In TGDH [1] every group member creates a key tree separately. Each leaf node is associated with a real group member, while each non-leaf node is considered as virtual member. In TGDH, every node on the key tree has a Diffie-Hellman key pair based on the prime p and generator α, used to generate the group key. Secret-public key pair for real member M_i is as follows.

$$\{KM_i, BKM_i = \alpha^{KM_i} \bmod p\} \quad (1)$$

And Secret-public key pair for virtual member V_i is as follows.

$$\{KV_i, BKV_i = \alpha^{KV_i} \bmod p\} \quad (2)$$

Public key BKM_i is also called as blinded key. Consider a node *Mv* whose left child is *Mlv* and right child node is *Mrv* (to simplify the description, we do not distinguish real members from virtual members here). Secret key of M_i's can be computed in the usual Diffie-Hellman key exchange fashions as follows.

$$\{KMv \equiv (BKMlv)^{KMrv} \equiv (BKMrv)^{KMlv} \bmod p\} \quad (3)$$

With all blinded keys well-known, each group member can compute the secret keys of all nodes on its key path, comprising the nodes from the leaf node up to the root. The root node's secret key KV_0 is known to all group members, and becomes the group key. In Figure 2, cluster member U_{12} knows the key pairs of U_{12}, V_{11} and V_{10}. V_{10}'s secret key is the cluster group key.

In AFTD [6], as increasing the group size, the group rekeying operation becomes complex and it leads to the performance degradation and generates more messages to distribute the group key, this is the main limitation of the AFTD protocol.

Renuka A. and K.C.Shet [9] were proposed the cluster based communications, which is different from our approach in key management and in communication protocols. Our detailed communication protocols and key management scheme are discussed in this paper.

Lee et al. [4,5] have designed several tree-based distributed key agreement protocols, reducing the rekeying complexity by performing interval based rekeying. They also present an authenticated key agreement protocol. As the success of their scheme is partially based on a certificate authority, their protocol will encounter the same problems as centralized trust mechanisms.

Nen-Chung Wang, Shian-Zhang Fang [10], have proposed 'A hierarchical key management scheme for secure group communications in mobile ad hoc networks'. This paper involves very complex process to form the cluster and for communications.

Gouda et al. [11], who describe a new use of key trees. They are concerned about using the existing subgroup keys in the key tree to securely multicast data to different subgroups within the group. Unlike their approach, which depends on a centralized key server to maintain the unique key tree and manage all keys, our paper solves this problem in a distributed fashion.

III. PROPOSED SCHEME

A. Sytem Model

To overcome the limitations of AFTD [6] protocol the entire set of group members in the network is divided into a number of subgroups called clusters and the layout of the network is as shown in Figure 1.

The cluster is formed based on location identification number, LID's of the users and clusters are assigned with cluster identification numbers, CID, which are given offline by the Certificate Authority CA. If the CID is equal to the LID then those users are belongs to that particular cluster.CID and LID are unique for each cluster.

In this paper each cluster member maintains its own cluster key tree as shown in Figure 2 (a,b,c), the leaf nodes in cluster key tree are the cluster users (real users), and non leaf nodes are the virtual users. We propose three different types of communication protocols with distributed tree-based group key management.

The cluster communications protocols are given below.

- Intra Cluster Communication protocol (ICC),
- Inter Cluster Communication protocol (IRCC) and
- Global Communication (GC) protocol.

Communication among the users within the cluster is called Intra Cluster Communication. Communication between the clusters is called Inter Cluster Communication. When IRCC occurs between the clusters then the respective cluster key tree is generated as shown in Figure 4, for generating group key. Communication among all clusters is called Global Communication and corresponding cluster key tree is generated as shown in Figure 5, for generating group key. The illustrations of communications are as shown in Figure 3.

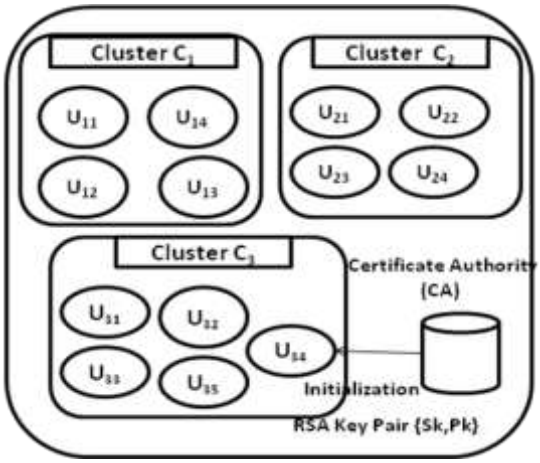
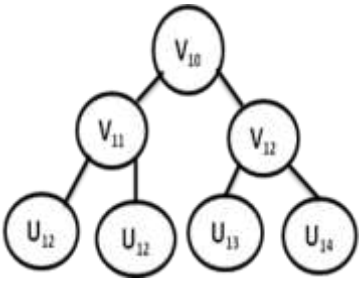
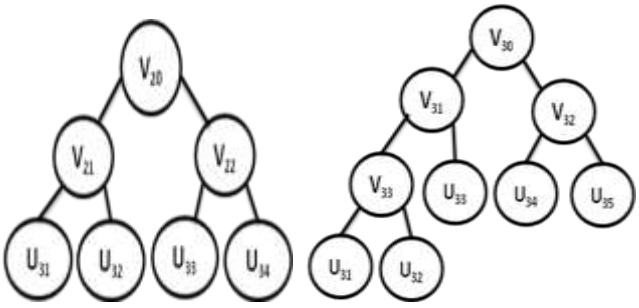


Figure 1. Network Layout and Initialization



a. Cluster C_1 Key Tree



b. Cluster C_2 Key Tree c. Cluster C_3 Key Tree

Figure 2. Key trees of clusters

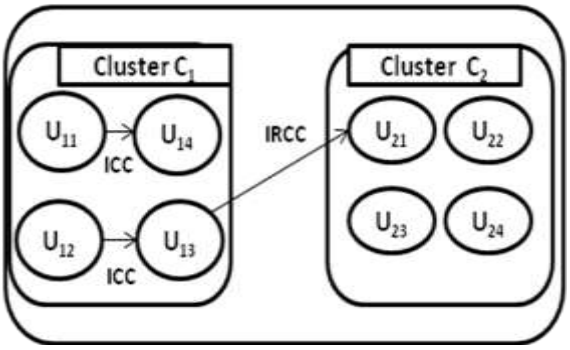


Figure 3. Illustration of communications.

B. Group Key Management Scheme

In fact an update of a blinded key need be sent only to a cluster members, instead of entire group (all clusters) based on the type of communications. We send each nodes blinded keys only to its cluster members. In this paper each cluster member constructs a key independently. Each real user U_{ij} of a cluster C_i has two key pairs first one is: Diffie-Hellman key pair, which is used to generate the group key is given below.

$$\{KU_{ij}, BKU_{ij} = \alpha^{KU_{ij}} \bmod p\} \quad (4)$$

And an RSA secret-public key pair {*Dij, Eij*}, which is used to provide source authentication. In key tree non-leaf nodes are virtual users (virtual clusters for global communication or for inter cluster communications), and have only a Diffie-Hellman key pair as given below.

$$\{KV_{ij}, BKV_{ij} = \alpha^{KV_{ij}} \bmod p\} \quad (5)$$

Group key management for user communications is occurs in two phases.

- Initialization phase
- Group key generation and distribution phase

1) Initialization Phase

Certificate authority, CA will distribute the appropriate public key certificates to clusters and it does not issue renewed public key certificates for existing group members during the process of cluster or group key updation.

New member wishing to join the group may obtain joining certificate and LID (based on location where user wants to join) from the CA at any time prior to join.

The certificate authority (CA), uses an RSA secret- public key pair {*Sk, Pk*} and establishes public key certificates for each cluster user U_{ij} by signing U_{ij}'s public key with its secret key *Sk*. User U_{ij}'s public key certificate $<U_{ij}, PUBU_{ij}, E_{ij} >Sk$ is now distributed to its cluster user since public key *Pk* is well known, any user of cluster can verify this certificate and obtains U_{ij}'s public key.

2) Group Key Generation and Distribution Phase

Group key generation and distribution for cluster communication occurs in three different ways.

- Group key generation and distribution in ICC.
- Group key generation and distribution in IRCC.
- Group key generation and distribution in GC.

The above group key generation and distribution techniques for cluster communications are implemented in respective communication protocols and in dynamic peer groups (in section 5).

IV. COMMUNICATION PROTOCOLS

The communication protocols are as follows.

- Intra Cluster Communications (ICC).
- Inter Cluster Communications (IRCC).
- Global Communications (GC).

A. Intra Cluster Communications(ICC)

Communication among the users within the cluster is called Intra Cluster Communication. Example of intra cluster communication is shown in Figure 3, and corresponding cluster key tree is shown in Figure 2.

In order to communicate users with each other within the cluster, they need to have the common cluster group key, which is generated from their cluster key trees based on Diffie-Hellman key exchange fashion.

Steps for generation and distribution of cluster group key in ICC (algorithm for cluster common key generation in ICC).

- Select the cluster in which Intra Cluster Communication is to be done.
- Each cluster (C_i) generates its own cluster key trees.
- The root node (V_{ij}) of cluster C_i's secret key KV_{ij} is generated using the DH Key exchange fashion from its leaf nodes (the generation of Cluster group key or common key is explained in dynamic peer groups).
- The secret key of the root node V_{ij} is KV_{ij} will become the cluster group key or common key for cluster C_i and that will be shared by all members of a cluster.
- For each session the cluster group key will be changed by changing their contribution.
- New generated cluster group key KV_{ij} will be distributed among all members of cluster.

B. Inter Cluster Communications(IRCC)

Communicating one cluster with another cluster is called an Inter Cluster Communication. The example of IRCC is shown in Figure 3, and corresponding reduced cluster key tree is generated as shown in Figure 4. In this figure VC0 is virtual cluster and it has only DH key pair as shown below.

$$\{KVC_i, BKVC_i = \alpha^{KVC_i} \bmod p\} \quad (6)$$

The secret-public key pair of virtual cluster VCi is for generating clusters common key, which is generated according DH Key fashion and distributed to the both clusters for communicating each other.

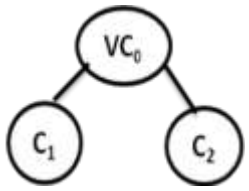


Figure 4. Reduced IRCC Key Tree.

The steps for Generation and distribution of common key for clusters in IRCC (algorithm for group key generation in IRCC)

- Select the clusters for IRCC and form reduced cluster key tree as shown in Figure 4.
- Each cluster has its own cluster group key or cluster's common key, which is generated from their cluster key tree based on DH key fashion.
- Cluster C_i and cluster C_j's secret keys KC_i, KC_j are calculated respectively (as explained in intra cluster communication algorithm).
- Using KC_i and KC_j, the root node VC_i (parent node of C_i and C_j, or virtual cluster) calculates its secret key KVC_i using DH key exchange fashion.
- The root node's VC_i is, *KVCi* which is common key for both cluster C_i and cluster C_j.
- KVC_i is distributed to both cluster and that will be shared by all members of each cluster for communicating each other.
- For each session the common key for clusters is recalculated by changing their shares of each clusters members and distributed to all members of both clusters.

C. Global Communication(GC)

Communicating all clusters in a group is called Global Communication. When cluster C1, C2 and C3 are communicating, then reduced global communication key tree is generated as shown in Figure 5, and common global key is generated according to DH key exchange fashion. In this figure leaf nodes are real clusters and non-leaf nodes are virtual clusters.

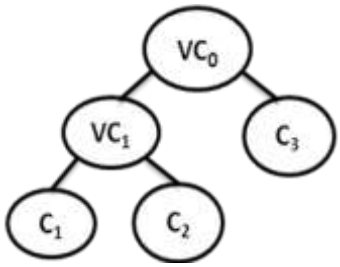


Figure 5. Reduced GC Key Tree

Steps for global key generation and distribution in GC (algorithm for global key generation & distribution in GC).

- Each cluster generates its own cluster key trees
- For each cluster key tree there will be generated the roots secret keys, which are common keys for all respective clusters.
- Cluster C_i, C_j and C_k's secret keys KC_i, KC_j and KC_k are calculated respectively from their cluster key trees.
- With these three clusters, Reduced Global Communication Key tree will be formed as shown in Figure 5.
- The root node VC_i's (from Reduced GC key tree) secret key KVC_i is calculated using DH key fashion, which is common key for all clusters C_i, C_j and C_k.
- VC_i's secret key KVC_i is distributed to all clusters and that will be shared by all members of each cluster for communicating globally.
- For each session the global key recalculated by changing their shares of each cluster's members and distributed to all members of clusters.

V. Dynamic Peer Groups

The numbers of nodes or clusters in the network are not necessarily fixed. New node (user) or cluster may join the network or existing nodes or cluster may leave the network.

A. User Joins the Cluster

Assume that a new user $U_{ij}+1$ wish to join a k-users cluster $\{U_1, U_2 \ldots U_K\}$. $U_{ij}+1$, is required to authenticate itself by presenting a join request signed with SK. $U_{ij}+1$ may obtain a signature on its join request by establishing credentials with the offline certificate authority.

When the users of clusters receive the joining request, they independently determine $U_{ij}+1$'s insertion node in the key tree, which is defined as in [1], which is the shallowest rightmost node or the root node when the key tree is well-balanced. They also independently determine a real user called join sponsor Us [1], to take responsible for coordinating the join, which is the rightmost leaf node in the sub tree rooted at the insertion node.

No keys change in the key tree at a join, except the blinded keys for nodes on the key path for the sponsor node. The sponsor simply re computes the cluster group key, and sends updates for blinded keys on its own key path to their corresponding clusters. The join works as shown below.

Steps for group key or cluster common key generation and distribution when user joins in cluster (algorithm for user joins in cluster).

- New User $U_{ij}+1$ takes the LID and public key certificates from the CA.
- User $U_{ij}+1$ selects appropriate cluster by comparing its LID with CID (for LID=CID).
- The user $U_{ij}+1$ broadcast the signed join request to its cluster C_i.
- Cluster C_i's members determine the insertion point, and update their key trees by creating a new intermediate node and promoting it to become the parent of the insertion node and $U_{ij}+1$.
- Each cluster member adjusts the cluster key tree by adding $U_{ij}+1$ to its selected clusters adjacent to the insertion point.
- The sponsor Us compute the new cluster group key or cluster common key.
- Then sponsor Us sends the updated blinded keys of nodes on its key path to their corresponding clusters.
- These messages are signed by the sponsor Us.
- $U_{ij}+1$ takes the public keys needed for generating the cluster group key, generates group key.

The cluster group key (for cluster C_3) or cluster common key for Figure 6 is generated as follows (steps for group key or common key generation).

- Let U_{31}'s secret share is $KU_{31,}$ and then secret-public key pair of U_{31} (according to DH Key fashion) is as shown below.

$$\{KU_{31}, BKU_{31} = \alpha^{KU_{31}} \bmod p\} \quad (7)$$

- Let U_{32}'s secret share is KU_{32} then secret-public key pair of U_{32} (according to DH Key fashion) is shown below.

$$\{KU_{32}, BKU_{32} = \alpha^{KU_{32}} \bmod p\} \quad (8)$$

- Let U_{33}'s secret share is KU_{33} then secret-public key pair of U_{33} (according to DH Key fashion) is shown below.

$$\{KU_{33}, BKU_{33} = \alpha^{KU_{33}} \bmod p\} \quad (9)$$

- Let U_{34}'s secret share is KU_{34} then secret-public key pair of U_{34} (according to DH Key fashion) is shown below.

$$\{KU_{34}, BKU_{34} = \alpha^{KU_{34}} \bmod p\} \quad (10)$$

- Let U_{35}'s secret share is KU_{35} then secret-public key pair of U_{35} (according to DH Key fashion) is shown below.

$$\{KU_{35}, BKU_{35} = \alpha^{KU_{35}} \bmod p\} \quad (11)$$

- Now V_{33}'s Secret-Public keys (KV_{33}, BKV_{33}) are calculated as follows (according to the DH Key Exchange fashion from U_{31} and U_{32}).

$$\{KV_{33} \equiv (BKU_{31})^{KU_{32}} \equiv (BKU_{32})^{KU_{31}} \bmod p\} \quad (12)$$

$$\{BKV_{33} = \alpha^{KV_{33}} \bmod p\} \quad (13)$$

- Now V_{32}'s Secret-Public keys (KV_{32}, BKV_{32}) are calculated as follows (according to the DH Key Exchange fashion from U_{34} and U_{35}).

$$\{KV_{32} \equiv (BKU_{34})^{KU_{35}} \equiv (BKU_{35})^{KU_{34}} \bmod p\} \quad (14)$$

$$\{BKV_{32} = \alpha^{KV_{32}} \bmod p\} \quad (15)$$

- Now V_{31}'s Secret-Public key pair (according to the DH Key Exchange fashion from V_{33} and U_{33}) is

$$\{KV_{31} \equiv (BKV_{33})^{KU_{33}} \equiv (BKU_{33})^{KV_{33}} \bmod p\} \quad (16)$$

$$\{BKV_{31} \equiv \alpha^{KV31} \bmod p\} \quad (17)$$

- Finally V_{30}'s Secret-Public key pair (according to the DH Key Exchange fashion from V_{31} and V_{32}) is

$$\{KV_{30} \equiv (BKV_{31})^{KV_{32}} \equiv (BKV_{32})^{KV_{31}} \bmod p\} \quad (18)$$

$$\{BKV_{30} = \alpha^{KV_{30}} \bmod p\} \quad (19)$$

- The root node V_{30}'s Secret key is considered as cluster C_3's Group key or cluster common key, through which communication is need to done.
- And this common cluster key is distributed to all cluster members.

Like above steps for group key or common key generation, the common key or group key for all the different cluster communication and in dynamic peer, are generated.

In Figure 6, a new user U_{36} wants to joins in C_3 cluster. The join sponsor U_{33} creates a new intermediate node V_{34} in the key tree and promotes it to become the parent of U_{33} and U_{36}. The sponsor U_{33} computes the new cluster group key, and sends the updated BKV_{34} and BKV_{31} to remaining members $\{U_{31}, U_{32}, U_{34}, U_{35}\}$ of the cluster C_3.

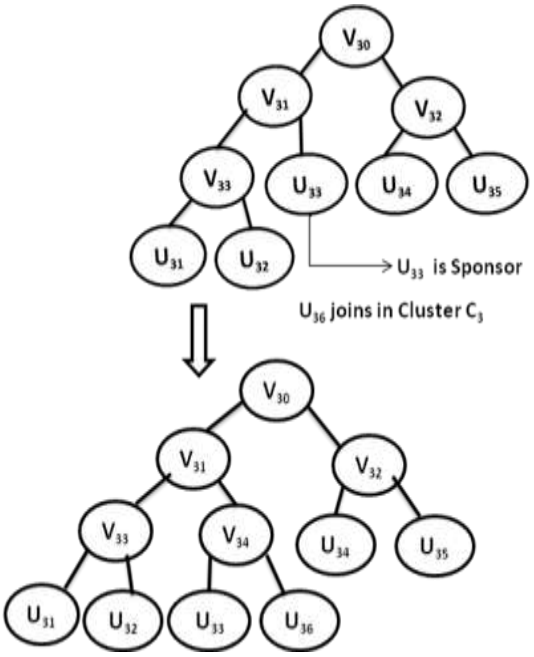


Figure 6. User joins in Cluster C3

B. User Leaves the Cluster

Assume that a member U_{ij} wishes to leave an n-member cluster. First U_{ij} initiates the leave protocol by sending a leave request. When the other users of cluster receive the request, they independently determine the sponsor node, which is the right-most leaf node of the Sub tree rooted at the leaving member's sibling node which is defined as in [1]. The leave protocol works as given below.

Steps for group key generation and distribution when user leaves the cluster (algorithm for user leave from cluster).

- User U_{ij} broadcasts its leave request to remaining users of that cluster C_i.
- The former sibling node of U_{ij} is promoted to replace U_{ij}'s parent node.
- The size of the cluster that formerly contained U_{ij} is decreased by one.
- The sponsor *Us* picks a new secret key *KUs*, and computes the new cluster group key, and sends the updated blinded keys of nodes on its key path to their corresponding cluster users.
- These messages are signed by the sponsor *Us*
- Group prepared based on DH key exchange fashion, as explained in dynamic peer groups.

In Figure 7, U_{36} leaves a cluster C_3. The sponsor U_{33} picks a new secret key KU_{33} and computes the new group key, sends updated BKU_{33}, BKV_{31} and BKV_{30} to their cluster users $\{U_{31}, U_{32}, U_{34}, and\ U_{35}\}$.

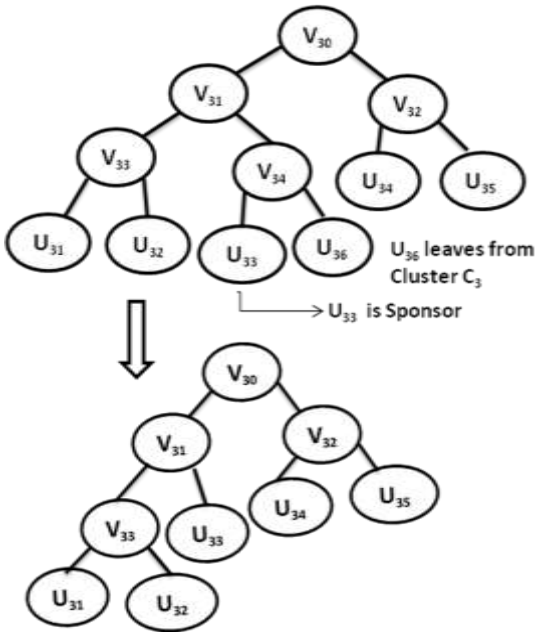


Figure 7. User U_{36} leaves the Cluster C_3.

C. Updating Secret Shares &RSA keys

In this scheme, each group user is required to update its Diffie-Hellman keys before each group session, or during a session when it is selected as a sponsor on a user's leaving. Source authentication of the updated blinded keys is guaranteed by the sender's RSA signature. Further, to ensure the long-term secrecy of the RSA keys, group user to renew its RSA key pair periodically, and send it to its cluster users securely using its current RSA secret key.

VI. Performance Analysis

Security Analysis: Users in a network group are usually considered to be part of the security issue since there are no fixed nodes to perform the service of authentication. The Certificate Authority, which may be distributed, is on-line during initialization, but remains offline subsequently. During initialization, the CA distributes key certificates and location IDs, so that the function of key authentication can be realized and distributed across appropriate clusters.

A. Forward Secrecy

If a hacker (or old member) can compromise any node and obtain its key, it is possible that the hacker can start new key agreement protocol by impersonating the compromised node. For our scheme we can conclude that a passive hacker who knows a contiguous subset of old group keys cannot discover any subsequent group key. In this way, forward secrecy can be achieved.

B. Backward Secrecy

A passive hacker (or new joined member) who knows a contiguous subset of group keys cannot discover how a previous group key is changed upon a group join or leave.

C. Key Independence

This is the strongest property of the dynamic peer groups. It guarantees that a passive adversary who knows some previous group key cannot determine new group keys.

VII. Conclusion

In this paper, we have presented three communication protocols with distributed group key management for dynamic peer groups using key trees, by dividing group into subgroups called clusters. We provided the strong authentication with LID's, CID's for cluster formations. We provide the source authentication of user in communication with RSA keys. The DH secret-public key pairs are used for common key generations. Certificate Authority provided the RSA keys, LID's for all users and CID's for all clusters for all types of cluster communications.

In future we can extend this application with cluster head communications, sponsor coordination and cluster merging or cluster disjoining in dynamic network.

Acknowledgment

We would like to thank to K Sahadeviah for help full discussion about different key management schemes and modes of providing authentications. We thank Krishna Prasad for discussion of effective presentations of concepts. We also thank our friends for designing of network frame work.

References

[1] Kim, Y., Perrig, A., Tsudik, G.: Simple and fault-tolerant key agreement for dynamic collaborative groups. In: Proceedings of the CCS'00. (2000).

[2] Steiner, M., Tsudik, G., Waidner, M.: Key agreement in dynamic peer groups. IEEE TRANSACTIONS on Parallel and Distributed Systems 11 (2000).

[3] Perrig, A.: Efficient collaborative key management protocols for secure automonomous group communication. In: Proceedings of CrypTEC'99. (1999).

[4] Lee, P., Lui, J., Yau, D.: Distributed collaborative key agreement protocols for dynamic peer groups. In: Proceedings of the ICNP'02. (2002).

[5] Lee, P., Lui, J., Yau, D.: Distributed collaborative key agreement protocols for dynamic peer groups. Technical report, Dept. of Computer Science and Engineering, Chinese University of Hong Kong (2002).

[6] Zhou, L., C.V.Ravishankar: Efficient, authenticated, and fault-tolerant key agreement for dynamic peer groups. Technical Report 88, Dept. of Computer Science and Engineering, University of California, Riverside (2004).

[7] Wong, C., Gouda, M., Lam, S.: Secure group communication using key graphs. In: Proceedings of the ACM SIGCOMM'98, Vancouver, Canada (1998).

[8] Steiner, M., Tsudik, G., Waidner, M.: Cliques: A new approach to group key agreement. In: Proceedings of the ICDCS'98, Amsterdam, Netherlands (1998).

[9] Renuka A. and K.C.Shet: Cluster Based Group Key Management in Mobile Ad hoc Networks (2009).

[10] Nen-Chung Wang, Shian-Zhang Fang.: A hierarchical key management scheme for secure group communications in mobile ad hoc networks (2007).

[11] M.G.Gouda, Huang, C., E.N.Elnozahy: Key trees and the security of interval multicast. In: Proceedings of the ICDCS'02, Vienna, Austria (2002).

[12] Wallner, D., Harder, E., Agee, R.: Key management for multicast: Issues and architecture. In: Internet Draft, draft-wallner-key-arch-01.txt. (1998).

[13] Ateniese, G., Steiner, M., Tsudik, G.: New multiparty authentication services and key agreement protocols. *IEEE* Journal of Selected Areas in Communications **18** (2000).

[14] Pereira, O., Quisquater, J.: A security analysis of the cliques protocols suites. In: Proceedings of the 14-th *IEEE* Computer Security Foundations Workshop. (2001).

[15] L.Zhou et Z. J. Haas, "Securing Ad Hoc Networks", *IEEE* Network Magazine, 13(6),1999.

[16] Eui-Nam Huh, Nahar Sultana, "Application Driven Cluster Based Group Key Management with Identifier in Mobile Wireless Sensor Networks(2007).

[17] D. Balenson, D. Mcgew, and A. Sherman, "Key management for large dynamic groups: One way function trees and amortized initializations," IETF, Feb 1999.

[18] Y.Kim,A. Perrig and G.Tsudik, " A common efficient group key agreement," Proc. IFTP-SEP 2001, pp,229-244,2001.

[19] Del Valle Torres Gerardo, Gomez Cardenas Roberto,"Overview the Key Management in Ad Hoc Networks (2004).

[20] Rafaeli, S. and Hutchison, D. (2003) A survey of key management for secure group communications, ACM Computing for secure group communication, ACM Computing Surveys, Vo. 35, No.3 pp.309-329.

[21] Bing Wu, Jie Wuand Yuhong Dong,(2008) An efficient group key management scheme for mobile ad hoc networks, Int. J. Security and Networks, 2008.

AUTHORS PROFILE

Mr. Rajendar Dharavath, currently is a Assistant Professor in the Department of Computer Science and Engineering, Aditya Engineering College, Kakinada, Andhra Pradesh, India. He completed B.Tech in CSE from CJITS Jangaon, Warangal, and M.Tech in CSE from JNTU Kakinada. His research interest includes: Mobile ad hoc networks, Network Security and Data Mining & Data Warehouse.

Mr. Bhima K, currently is a Associate Professor and Head of Department of Computer Science and Engineering,Brilliant Institute of Engineering and Technology, Hyderabad, Andhra Pradesh, India. He completed B.Tech in CSE from RVR&JC Engg. College, Guntur, and M.Tech in SE from NIT Alahabad. His research interest includes: Mobile ad hoc networks, Network Security, Computer Networks and Software Engineering.