Communication in Vibrant Peer Groups For Cluster Key Management

Hemanth Siramdasu[1], Hari Krishna[2]

[1]CSE Department, VelTech Multitech Dr.Rangarajan Dr.Sakunthala Engineering college Chennai, India

[2]Tata Consultancy Services, Chennai, India

***Abstract*— In this paper we present an efficient way of group communication using key trees. group member can be formed as a tree structure and it is divided into global cluster and local clusters and group communication can be done in 3 different ways as follows 1) intra cluster communication 2) inter cluster communication 3) global communication. We also provide authenticated group key management in all above 3 cases of group communications. And we are going to reduce the number of rekeying operations and the numbers of group key distributions operations in the case of inter cluster communication and intra cluster communication and proving the flexible communication among the group members. Our approach is by insight that when a Diffie-Hellman public key is updated, in a cluster based method it suffices to update one cluster to other cluster which are going to be communicate based on type of communication. More over TA guaranteeing key authentication, enhancing the fault tolerance and protecting our protocol from all types of attacks. And we are going to proposing to reduce the communication overhead and increasing the performance.**

***Keywords*— Intra cluster communication & Inter cluster communication, AFTD protocol.**

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. Consequently not only does the group communication need to be secure but also efficient and flexible among group members as per the requirement. Research efforts have been put into the design of a group key management scheme for the sake of scalability, reliability, and security. Furthermore, group key management also needs to address the security issue related to membership changes. The modification of membership requires refreshment of the group key. This can be done either by periodic rekeying or updating right after member change. The change of group key ensures backward and forward security.

Group key management protocols can be roughly classified into three categories; centralized, decentralized, and distributed [1]. In centralized group key protocols, a single entity is employed to control the whole group and is responsible for group rekeying. In the decentralized approach, multiple entities are responsible for managing the group as opposed to a single entity. In the distributed method, group members themselves [2] contribute to the formation of a group key and are equally responsible for the rekeying and distribution of group keys. We propose a distributed group key management approach wherein there is no central authority and uses a combination of public and private key cryptography. We treat the total network topology as a group which forms binary tree structure and this tree divided into cluster and sub cluster. We assume in every cluster, every node can [3] receive a message broadcasted from the other nodes.. Each cluster is headed by a cluster head which is parent of users. The nodes within the cluster communicate using symmetric cryptography with the cluster group key. cluster group key is shared by all the cluster members[4]. Asymmetric key cryptography is used to encrypt the group keys generated, whenever membership changes occur. The authentication is done using Trusted Authority[10] by issuing the public key certificate prior to the time of joining the cluster or group.

To achieve the group communication common method is to encrypt the messages with a group key so that entities outside[5] the group member cannot decode them. In this paper we focus on flexible communication among the group members according the type of communication and group key management with authentication/integrity. This ensures that public keys of group members cannot be modified by adversaries.

Group key is updated on every membership change for forward secrecy and backward secrecy, a new method called group rekeying. To reduce the number of rekeying operations, Woung.et al proposed a logical data structure called a key tree that reduces the rekeying overhead from o(n) to o(nlogn) where n is the group size. And Kim et al proposed a tree-based key agreement protocol, TGDH which is combination of key tree and Diffie-Hellman key exchange to generate and maintain the group key. But it suffers from the impersonation attack and communication overhead. 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) which is the combination of key trees, Diffie-Hellman key exchange for group key generation and RSA key for strong authentication.

In AFTD protocol, key tree is based on the clusters. Cluster contains at most 3 members and at least 2 members[7]. If cluster size is multiple of two then the cluster is divided into two clusters and communication can be done only globally but Unfortunately AFTD suffers from the following drawback

a). there is no intra cluster communication and

b). there is no inter cluster communication

There is only the global communication

A. Our work:

In this paper we propose three different types of communication protocols using key trees. Here the group member forms as a tree structure and this tree is divided into global cluster, subcultures and group communication exist as follows

a) Communication within the cluster or intra cluster communication

b) Communication between the clusters or inter cluster communications

c) Global communications

In intra cluster communication there should be a common key within the cluster only and the cluster head key will become the group key. This group key will be generated from contribution of cluster members

In inter cluster communication, one cluster communicates with another cluster and common group key will be generated for clusters which are participating in communication. In global communication group key will be generated from root of the key tree similarly to the AFTD protocol

And proposed the following group key agreement strategies

i. Group key agreement in intra cluster communication

ii. Group key agreement in inter cluster communication

iii. Group key agreement in global communication

We achieve the group key generation according to the AFTD protocol.

When user U_i joins or leave the cluster group member then corresponding cluster sponsor will send updations of public key to its cluster by g casting messages for recomputing the group key.

II. RELATED WORK

Key trees were first proposed for centralized key distribution, while Kim et al. adapted it to distributed key agreement protocol TGDH. In TGDH every group member creates a key tree separately. Each leaf node is associated with a *real* group member, while each non-leaf node to a subgroup of the group *G*, considered a *virtual member*. In Figure 1, virtual member *V*4 corresponds to the subgroup that contains two real group members *M*3 and *M*4. 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[8][9]. Secret-public key pair $\{KMi, BKMi = \alpha KMi \bmod p\}$ is for real member *Mi*, and $\{KVi, BKVi = \alpha KVi \bmod p\}$ is for virtual member *Vi*. Public key *BKMi* is also called as *blinded key*. And latter it is used in AFTD protocol (authenticated fault-tolerant tree based deffie-hellman key exchange protocol) in which every member create a key tree separately each leaf node is associated with group user while each non leaf node is considered as virtual group user. In this protocol each group member constructs a key independently.& do not distinguish real members from virtual members here). *Mi*'s secret key can be computed in the usual Diffie-Hellman fashion as $KMv \equiv (BKlv)Krv \equiv (BKrv)Klv \bmod p$.
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 3, group member *M*2 knows the key pairs of *M*2, *V*3, *V*1 and *V*0. *V*0's secret key is the group key.

In AFTD, key tree is formed by the group members and latter group members divide into the clusters. Each cluster contains at most 3 members and at least 2 members. If cluster size is multiple of two then the cluster is divided into two clusters and communication can be done only globally

B. Limitations of the AFTD protocol:

a. There is no intra cluster communications if required

b. There is no inter cluster communications

c. There is only the global communications

d. for each and every type of communication they need to update and distribute public key to all .members, in that

case unnecessarily increasing the overhead of updating and distributing the group key to all members.

III.PROPOSED SCHEME

A. System model:

To overcome the above limitations of AFTD protocol and we need to divide binary tree into the global cluster, and sub cluster. In this paper group user forms binary key tree structure as shown in fig.2 and group members divided into cluster and sub clusters as shown in fig.3. And we are going to propose different types of communication protocols and also providing group key management in different type of communications.

The following are the different types of communication protocols

1. Intra cluster communications
2. Inter cluster communications
3. Global communications

Intra cluster communications: communications between the members of one cluster or

Communication within the cluster members

Example: communication exist between U_1 ,U_2 ,U_3 and U_4 or U_5 ,U_6 ,U_7 and U_8 .

Inter cluster communications: communication exists between any clusters

C_1, and C_2 or C_3 and C_4.

Global communications: communications among all group members

Ex: conferences

In order to overcome the above limitations in AFTD protocol , we need to define or divide the key tree into clusters and sub clusters

Initially group member forms as a tree structure as shown fig2

Latter it is divided into clusters as shown in fig3.

Tree figure which is divided into cluster and sub clusters.

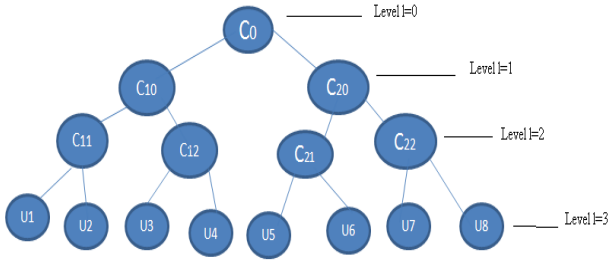


Fig.1

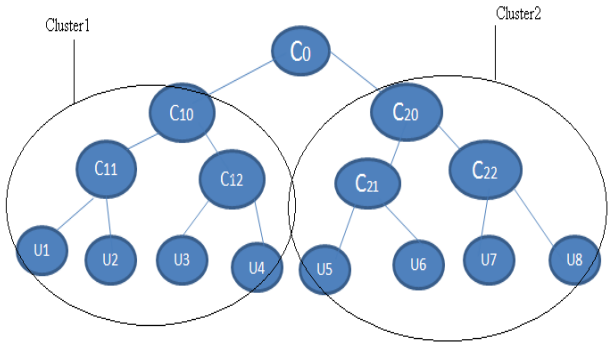


Fig.2

Cluster tree is shown below in fig.4

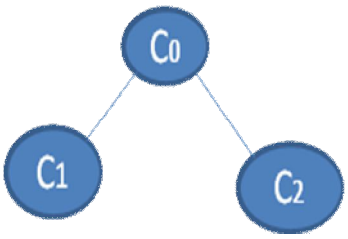


Fig.3

B. Advantages of cluster communications:

i. Communications can be done flexible as per our requirements

ii. If intra cluster communication done, then only that cluster member should have the common group key and this group key need not send or update to the all group members

iii. In the case of inter cluster communication, the common group key should be for cluster which are want to communicate each other, there is no need to send updations to all remain group

members remain cluster which are not participating in communications

iv. In the case of intra and inter cluster communications we can avoid the unnecessary of group key updating operations and distributions operations.

In above all 3 cases of communications group key agreement need to be done successfully.

First determine the type of communications and then follow the corresponding key agreement protocols.

IV. GROUP KEY MANAGEMENT PHASE:

In fact an update of a blinded key need be sent only to a cluster group instead of entire group based on the type of communications. We send each nodes blinded keys only to its cluster members. In this paper each cluster group member constructs a key independently. Each real group member of cluster Ui has two key pairs first one is: Diffie-Hellman key pair { KUi ,BKUi = αKUi mod p}, which is used to generate the group key, and an RSA secret-public key pair, {Di, Ei}, which is used to p provide source authentication. Non-leaf nodes are virtual members, and have only a Diffie-Hellman key pair {KCij ,BKCij = αKCij mod p}.

Group key management occurs in two phases

i. Initialization phase

ii. Group key generation and distribution phase

A. Initialization phase:

Trusted authority (TA) will distribute the appropriate public key certificates to clusters . TA will not issue renewed public key certificates for existing group members during the process of group key updations. New member wishing to join the group may obtain joining certificate from the TA at any time prior to join

In fig4. This trusted authority uses an RSA secret public key pair (Sk, Pk) and establishes public key certificates for each group member Ui by signing Ui's public key with its secret key Sk. Ui's public key certificate <Ui, PUB_{Ui},Ei >Sk is now distributed to its cluster member since public key Pk is well known, any member of cluster can verify this certificate and obtains Ui's public key.

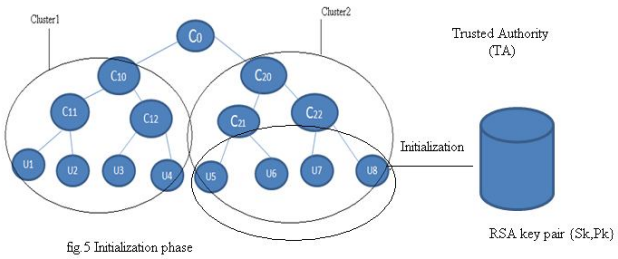


fig.5 Initialization phase

Group key generation and distribution occurs in 3 different ways

1. Group key generation and distribution in intra cluster communication
2. Group key generation and distribution in inter cluster communication
3. Group key generation and distribution in global communication.

V. COMMUNICATION PROTOCOLS:

1. Intra cluster communications
2. Inter cluster communications
3. Global communications

*1) Intra cluster communications***:** Communications among the member s of one cluster or Communication within the cluster members

For example communications within C1 (or C2) cluster members Generation of group key in intra cluster communication

a) C10 or C20 calculates it's secret key KC_{10} or (KC_{20}) in a DH \Key exchange fashion.

b) The secret key of the C_{10} or C_{20} is KC_{10} (or KC_{20}) will become the cluster group key of C1(or C2) and that will be shared by all group members within cluster

c) For each session the cluster group keys will be changed by changing their contribution. And changed cluster group key will be distributed among all members of clusters.

2) Inter cluster communications: Communicating one cluster with another cluster is called as inter cluster communication. For example cluster C1 with C2 (in this paper we have taken two clusters hence inter cluster communication will be same as global communications for differentiating inter cluster communication and global communication we need to have more than two cluster)

Generation of group key in inter cluster communication

a) Each member in cluster maintains its cluster key tree

b) Cluster C1 and cluster C2's secret keys are KC_{10}, KC_{20} respectively. with these two keys C_0 calculates its secret key KC_0 using DH key fashion, which is common key for both cluster C1 and C2.

c) C_0's secret key KC_0 is distributed to both cluster and that will be shared by all members of each cluster for communicating each other

d) For each session the common group key recalculated by changing their shares of each clusters members and distributed to all members of clusters which are in inter cluster communication.

VI. DYNAMIC PEER GROUPS

The number of nodes in the network is not necessarily fixed. New nodes may join the network or existing nodes may leave the network.

*1)New member joins the cluster***:** Assume that a new member Ui+1 wishes to join a k-member cluster which smaller than other clusters, and contains *{U*1,*U*2, . . . ,*Uk }* . Ui+1 is required to authenticate itself by presenting a join request signed with *SK*. UI+1 may obtain a signature on its join request by establishing credentials with the offline trusted authority,. When the other cluster members receive this request, they independently determine Ui+1's insertion node [1] in the key tree, which is the shallowest rightmost node, or the root node when the key tree is well-balanced. They also independently determine a real member called *join sponsor Us* 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 recomputes the group key, and sends updates for blinded keys on its own key path to their corresponding clusters. The join works as shown in Algorithm 1.

Algorithm 1 Join Protocol

1: Compare cluster C1 and C2, if C1>C2 then new user joins in C2 otherwise joins in C1 (if C1=C2 then new user joins in C1)

2:The new user Ui+1 broadcasts the signed join request to the cluster C2.

2: cluster C2'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 both the insertion node and Ui+1.

3: Each cluster member adjusts the cluster key tree by adding Ui+1 to the smallest cluster adjacent to the insertion point,

4: The sponsor Us computes the new group key, and sends the updated blinded keys of nodes on its key path to their corresponding clusters. These messages are signed by the sponsor Us.

7: Ui+1 the public keys needed for generating the group key.

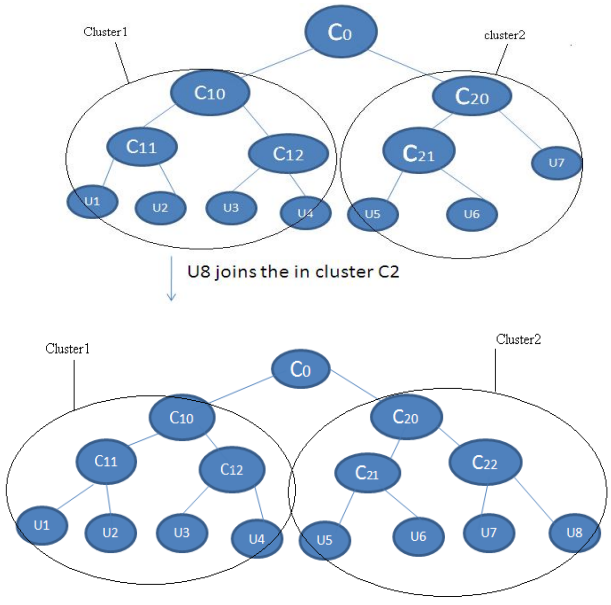


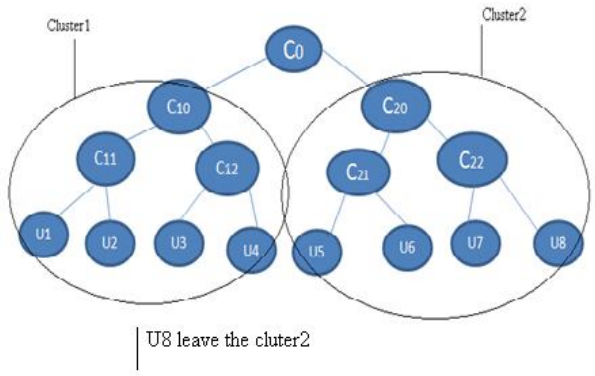
Fig.5

In Figure 5, C1cluster > C2 cluster hence new user U8 joins in C2 cluster . The join sponsor U7 creates a new intermediate nodeC_{22} in the key tree and promotes it to become the parent of U7 and U8. The sponsor U7 computes the new group key, sending the updated BKC_{22} and BKC_{20} to remaining members {U5,U6,U7,U8 } of the cluster C2.

2) Old member leave the group: Assume that a member U_L wishes to leave a *n*-member cluster. First U_L initiates the leave protocol by sending a leave request. When the other group members receive the request, they independently determine the sponsor node, which is defined as in to be the right-most leaf node of the Sub tree rooted at the leaving member's sibling node. The leave protocol works as shown in Algorithm 2.

Algorithm 2 Leave Protocol
1: The former sibling node of U_L is promoted to replace U_L's parent node.
2: The size of the cluster that formerly contained U_L is decreased by one,
3: The sponsor Us picks a new secret key *KUs* , computes the new group key, and sends the
updated blinded keys of nodes on its key path to their corresponding cluster members. These messages are signed by the sponsor Us.

In Figure 6, U8 leaves a cluster C2. The sponsor U_7 picks a new secret key KU_7 and computes the new group key, sending updated BKU_7 , BKC_{20} to their cluster members $\{U5,U6,U7,\}$.



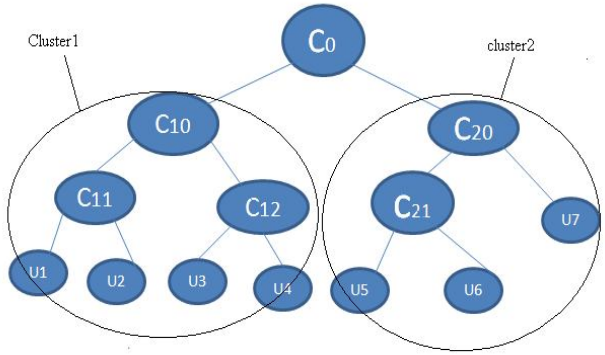


Fig.6

Updating Secret Keys &RSA keys:
In this scheme, each group member 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 member'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 member to renew its RSA key pair periodically, and send it to its cluster members securely using its current RSA secret key.

VII. PERFORMANCE ANALYSIS :

1)Security analysis: Members 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 trusted authority, which may be distributed, is on-line during initialization, but remains offline subsequently. During initialization, the TA distributes valid key certificates, so that the function of key authentication can be realized and distributed across appropriate clusters. Since the duration of initialization is relatively short, it is safe for us to use the TA at that time.

2)Group Key Secrecy **:** This is the most basic property. It guarantees that it is computationally infeasible for a passive adversary to discover any group key.

3)Forward Secrecy : When a new member joins we ensure that it is not able to receive the previous information that was exchanged prior to it joining the network.

*4)Backward Secrecy***:** Guarantees that a passive adversary who knows a contiguous subset group keys cannot discover preceding group keys.

5)Key Independence : The strongest property. It guarantees that a passive adversary who knows a some previous group key cannot determine new group keys

6)Computation Overhead and Storage Requirements

i. Depending on the type of communication the storage capacity will be varies.

ii. As increase the height of key tree storage over head will be decreased.

CONCLUSION

In this paper, we have presented an efficient, way of communications in dynamic peer groups by dividing group into cluster and sub cluster and generated group key in cluster is authenticated and fault tolerant which is based on tree-based key agreement protocol. Our performance analysis shows that our approach can significantly reduce the communication and storage overheads.

REFERENCES

[1] H. M. N. D. Bandara and A. P. Jayasumana, "Collaborative applications over peer-to-peer systems - Challenges and solutions," Peer-to-Peer Networking and Applications, Springer, 2012

[2] Chi-Jyh Guo, Yuh-Ming Huang, "*Residency based Distributed Collaboration key Agreement for Dynamic Peer Groups"*International Journal of Innovative Computing Information & Control, vol.8,No.8,Aug.2012.

[3] Sandro Rafeali, David Hutchison, "A survey of key management for secure group communication " ACM Computing Surveys, vol.35, iss3,Sep.2003.

[4] Lee, P.P.C, "*Distributed collaborative key agreement and authentication protocols for dynamic peer Group*" IEEE Transactions on Networking, vol.14,iss.6,April2012.

[5] P. Lee, A. P. Jayasumana, H. M. N. D. Bandara, SLim, and V. Chandrasekar, "A peer-to-peer collaboration framework for multi-sensor data fusion," J. of Network and Computer Applications, vol. 35, no. 3, May 2012, pp. 1052–1066.

[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 (2003)

[7] Hanaa Torkey,Gamal Atiya, "*Modified Fast Recovery Algorithm for Performance Enhancement of TCP-NewReno*"IJCA vol.40,iss.12, Feb 2012..

[8] M. S. Artigas and B. Herrera, " *SocialHelpers: Introducing social trust to ameliorate churn in P2P reputation systems," In Proc. IEEE Int. Conf. on Peer-to-Peer Computing*", Aug.-Sep. 2011, pp. 328–337.

[9] H. M. N. D. Bandara and A. P. Jayasumana, "*Evaluation of P2P resource discovery architectures using real-life multi-attribute resource and query characteristics*," In Proc. IEEE Consumer Communications and Networking Conf. (CCNC '12), Jan. 2012.

[10] P. Ganesan, B. Yang, and H. Garcia-Molina, "*One torus to rule them all: Multi-dimensional queries in P2P systems*," In Proc. 7th Int. Workshop on Web and Databases (WebDB '04), June 2004.