ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 2, February 2019

Secure Group Key Agreement Algorithm for Mobile Adhoc Networks

Gurudeepa G[1], Dr.S.Amutha, M.E., Ph.D., [2]

PG Scholar, Department of Computer Science and Engineering, PSR Engineering College, Sivakasi, Tamilnadu, India[1]

Associate Professor, Department of Computer Science and Engineering, PSR Engineering College, Sivakasi, Tamilnadu, India[2]

ABSTRACT: Social network plays a major role these days amongst users. Social network is a concept to connect people and to share personal details with their closest ones and these applications supporting group activities. Key agreement is suitable for the ad-hoc network because it does not need a TTP (Trusted Third Party). Group key agreement time means the time spent to share a group key between all members. To reduce the time, the tree-based key agreement method has been proposed and many protocols use this idea. We propose a group key agreement algorithm where the group members have different capabilities. This protocol has the goal of reducing the group key agreement time. We propose to gain a shorter group key agreement time by substituting the Generalization Diffie-Hellman (GDH) key agreement computations with Arbitrary Topological computations using symmetric key. A Secure hash algorithm provides a framework for offering the services needed to supply security over a network.

KEYWORDS: Group key, Key agreement algorithms, Secure hash algorithm, Symmetric key

I. INTRODUCTION

Social network is the concept to connect people and to share personal details to their closest ones. A social network consists of, a set of nodes and there exist a relation between the existing nodes. The node may either individual or group. A group of users they want to communicate securely in the encryption and decryption format so we go for the group key agreement scheme for social network group users. The group key management protocols can be generally classified as Network independent based and Network dependent based key management protocols. The network independent based key management protocol is further classified into centralized, decentralized and distributed key management protocols, whereas the network dependent based key management is classified as tree based and cluster based key management.

The role of key management also extends to a participating member authentication, to prevent any intruder from impersonating and providing access control to validate the joining operation. Moreover, the key management follows a set of cryptographic techniques in generating and distributing the keys which may be symmetric or asymmetric for secure group communication.

II. RELATED WORK

Shaoquan Jian[1] proposed an group key agreement with an arbitrary connectivity graph, where each user is only aware of his neighbours and has no information about the existence of other users. Further, user has no information about the network topology. Under this setting, a user does not need to trust a user who is not his neighbour. Thus, if one is initialized using PKI, then he need not trust or remember public-keys of users beyond his neighbours. To update the group key more efficiently than just running the protocol again, when user memberships are changing. Two passively secure protocols with contributiveness and proved lower bounds on a round complexity, demonstrating that the protocols are round efficient.

E.Bresson, O.Chevassut, and D. Pointcheval [2] proposed Authenticated Diffie-Hellman Key exchange allows two principals communicating over a public network, and each holding public/private keys, to agree on a shared secret value. In this paper the natural extension of this cryptographic problem to a group of principals .it begin existing formal security models and refine them to incorporate major missing details. Within this model we define the execution of a protocol for authenticated dynamic group Diffie- Hellman and show that it is provably secure under the decisional Diffie-Hellman assumption. Our security result holds in the standard model and thus provides better security guarantees than previously published results in the random oracle model. Implementation Methodologies used in this paper are Decisional Diffie Hellman. Considers important attributes such us strong-corruption, concurrent executions of the protocol, tighter reduction to the group Diffie-Hellman key exchange. It has a well-defined security model by considering the Multi Diffie- Hellman (M-DH) and Random Multi Diffie-Hellmandistribution.

M. Burmester and Y. Desmedt[3] proposed practical conference key distribution systems based on public keys, which authenticate the users and which are 'proven' secure provided the Diffie- Hellman problem is intractable. A certain number of interactions is needed but the overall cost is low. There is a complexity tradeoff. Depending on the network used, we either have a constant (in the number of conference participants) number of rounds (exchanges) or a constant communication and computation overhead. Our technique for authentication can be extended and used as the basis for an authentication scheme which is (proven' secure against any type of attack, provided the Discrete Logarithm problem is intractable. The system uses cyclic functions. That prevents the attack by passive eavesdroppers and maintaining the efficiency. The system protect any type of attacks including adaptive chosen so that attack by real-time middle- persons, under the same cryptographic assumption.

K. Yongdae, P. Adrian, and G. Tsudik[4] proposed group-oriented applications and protocols have been gaining popularity. Such applications typically involve communication over open networks where security is an important concern. Group key management is one of the basic building blocks in securing group communication. Most prior research in group key management focused on minimizing computation overhead due mostly to expensive cryptographic operations. Communication cost has been treated as a secondary concern. This has been (and perhaps still is) a reasonable strategy, however, certain changes are looming on thehorizon.

X. Lv, H. Li, and B. Wang[5] proposed on Self-organizing group key agreement protocols without a centralized administrator are essential to secure group communication in dynamic peer systems. In this paper, we propose a generic construction of a one-round self- organizing group key agreement protocol based on the Chinese Remainder Theorem. In the proposed construction, all group members contribute their own public keys to negotiate a shared encryption public key, which corresponds to all different decryption keys. Using his/her own secret key, each group member is able to decrypt any cipher text encrypted by the shared encryption key. Following the generic construction, one-round self-organizing group key agreement protocol using the efficient and computationally inexpensive public key cryptosystem NTRU. Both the public key and the message in this protocol are secure against the known lattice attacks.

Yang Yang [6] proposed a Mobile Ad-hoc Networks (MANETs) are considered as the most promising terminal networks in future wireless communications and characterized by flexibility, fast and easy deployment, which make them an interesting technology for various applications. Group communication is one of the main concerns in MANETs. To provide the secure group communication in wireless networks, a group key is required so that efficient symmetric encryption can be performed. In this paper, we propose a constant-round group key agreement scheme to enable securegroup communications, which adopts the Identity Based Broadcast Encryption (IBBE) methodology.

Emmanuel Bresson [7] proposed on Dynamic groupDiffie- Hellman protocols for Authenticated Key Exchange (AKE) are designed to work in a scenario in which the group membership is not known in advance but where parties may join and may also leave the multicast group at any given time. While several schemes have been proposed to deal with this scenario no formal treatment for this cryptographic problem has ever been suggested. In this paper, they done a security model for this problem and use it to precisely done Authenticated Key Exchange (AKE) with implicit" authentication as the fundamental goal, and the entity-authentication goal aswell.

Qianhong Wu [10] proposed on A group key agreement (GKA) protocol allows a set of users to establish a common secret via open networks. Observing that a major goal of GKAs for most applications is to establish a confidential channel among group members, we revisit the group key agreement definition and distinguish the conventional (symmetric) group key agreement from asymmetric group key agreement (ASGKA) protocols. Using bilinear pairings, we realize an efficient ASBB scheme equipped with useful properties. Following the generic construction, we instantiate a one-round ASGKA protocol tightly reduced to the decision Bilinear Diffie-Hellman Exponentiation (BDHE) assumption in the standard model.

III. PROPOSED WORK

The proposed system for optimal group key agreement algorithm for social network can be done by four different compositions .they are asfollows

1. Groupformation
2. Public keygeneration
3. Secret keygeneration
4. Group keygeneration

3.1 Architecture Design of Proposed System

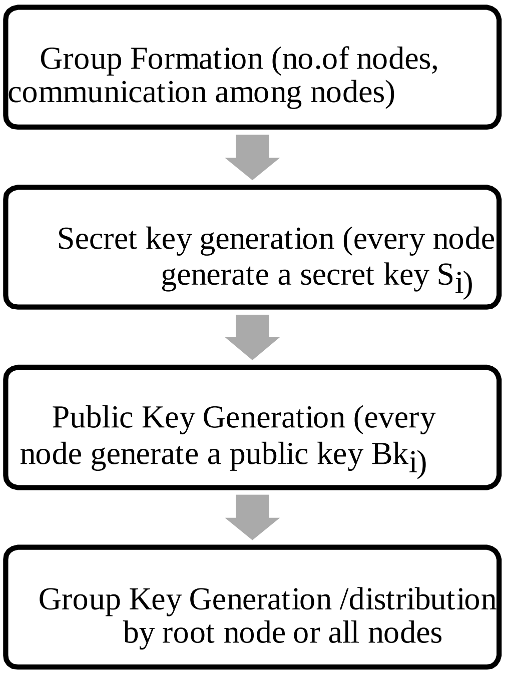
Group Formation (no.of nodes, communication among nodes)
Secret key generation (every node generate a secret key S_i)
Public Key Generation (every node generate a public key Bk_i)
Group Key Generation /distribution by root node or all nodes

Fig.3.1: Architecture design of proposed System

3.2 GROUPFORMATION

Proposed work to form a group using of key trees in fully distributed contributory key agreement. Number nodes are depend upon the user maximum weight of the node will be root node. the root is located at level 0 and the lowest level 0 and the lowest level a are level h. each node represented (l,v).each node(l,v) is associated with the key S(l,v) and the public key(bkey) Bk(l,v)=f(S(l,v)) where the function f() is modular exponentiation in prime order groups,i.e.f(s)=α^S mod p(analogous to theDiffiehellman protocol).assuming a leaf node(l,v) hosts the user U_i,the node(l,v) has U_i's session random key S(l,v).furthermore, the member U_i at node (l,v) knows every key along path form(l,v) to (0,0),referred to as the key- path denoted key_i..In figure 1,if a user U_2 owns the tree then U_2 knows every key{S(3,1),S(2,0),S(1,0),S(0,0)} .and every public key Bk_2={Bk(0,0),Bk(1,0),. Bk(l,v)} on tree. Every key S(l,v) is computed as follows S(l,v)=f(K(l+1,2v)K(l+1,2v+1))

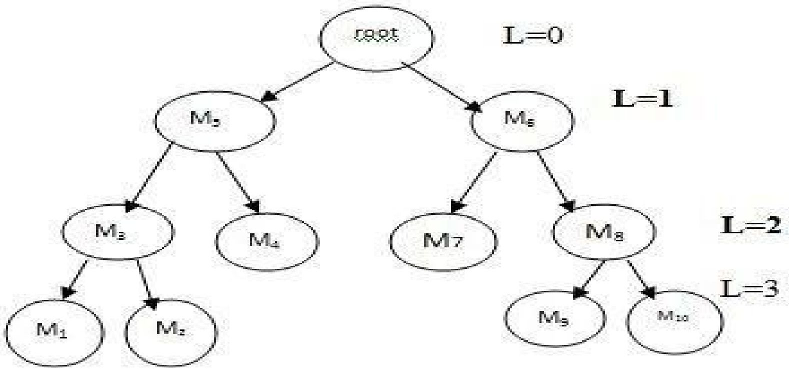

Fig: 3.2 Group formation tree

3.2.1 BASIC TREEOPERATION

In this section we introduce the four basic operation: join, leave, merge, and partition. All proto share a common framework with the following notable features:

– Each group member contributes an equal share to the group key. The key is computed as a function of all current group members‘shares.

– Each share is secret (private to each group member) and is never revealed.

– As the group grows, new members' shares are factored into the group key and, upon each new member's joining, one of the old members changes itsshare.

– As the group shrinks, departing members' shares are removed from the new group key, and at least one remaining member changes itskeyshare4.

– All protocol messages are signed, time stamped, sequence-numbered and type-identified by the sender. (We use Diffie-Hellman for message signing since the number of receivers is greater than the number ofsenders.)

A group key can be computed from any member's secret share (i.e., any leaf value) and all bkeys on the co-path to the root. It is easy to see that knowledge of its own secret share and all sibling bkeys on the path to the root enables a member to compute all intermediate keys on its key- path, including the root group key. This is similar to other tree based schemes where each member is required to know all keys on the path from itself (leaf) to the root. Although not strictly necessary for computing group key, our protocol also requires each member to know **all** bkeys in the entire key tree. As will be seen below, this makes the handling of future membership changes more efficient and robust. As part of the protocol, a group member can take on a special **sponsor** role which involves computing intermediate keys and broadcasting to the group.

Each broadcasted message contains the sender's view of the key tree which contains each bkey known to the sender. Any member in the group can unilaterally take on this responsibility, depending on the type of membership event. In some cases, such as a partition event, multiple sponsors might be involved. In case of an additive change (join or merge), all group members identify a unique sponsor. This sponsor is responsible for updating its secret key share, computing affected (S(l,v),Bk(l,v)) pairs and broadcasting all bkeys of the new tree to the rest of the group. The common criteria for sponsor selection is determined by the tree maintenance strategy.

We emphasize, from the outset, that sponsor is not a privileged entity: its only task is the updating and broadcasting of tree information to the group. In response to a subtractive membership change (leave or partition), all members update the tree in the same manner. Since the case of partition subsumes the case of a single leave, we discuss it in more detail. Group partition results in a smaller tree since some leaf nodesdisappear.

As a result, some sub trees acquire new siblings; therefore, new intermediate keys and bkeys must be computed through a Diffie-Hellman exchange between the new siblings sub-trees. The computation proceeds in a bottom-up fashion with each member computing keys and bkeys until either: it blocks due to a dependency on a new sibling bkey that it does not yet know, or it computes the new root (group)key.

If a member blocks without computing any new keys, it does nothing. Otherwise, it broadcasts its view of the key tree which includes the newly computed b keys. This process is repeated at most F times where F is the height of the tree, i.e., until all remaining members compute the new groupkey.

Before turning our attention to the actual protocols we stress that, while a comprehensive protocol suite must address all types of key adjustment operations, the general policy (or case-by-case decisions) regarding if and when to change a group key is the responsibility of the application and/or the group communication system. In this section we discuss only join and leave operation.

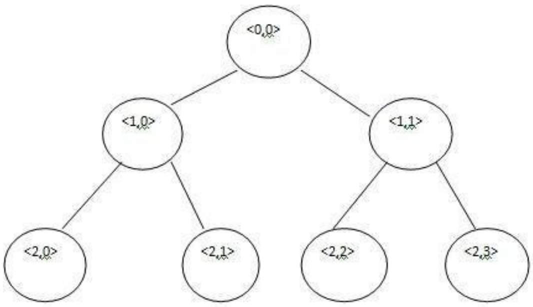

Fig3.3: Tree Operation with notations

3.2.2 JOINOPERATION

We assume the group has User {U1,....,Un} The new member Un+1 initiates the protocol by broadcasting a join request message that contains its own bkey (Bk(0,0)). This message is distinct from any JOIN messages generated by the underlying group communication system, although, in practice, the two might be combined for efficiency's sake. Each current member receives this message and determines the insertion point in the tree. The insertion point is the shallowest rightmost node, where the join does not increase the height of the key tree.

Otherwise, if the key tree is fully balanced, the new member joins to the root node. The sponsor is the rightmost leaf in the sub tree rooted at the insertion node. Next, each member creates a new intermediate node and a new member node, and promotes the new intermediate node to be the parent of both the insertion node and the new member node. After updating the tree, all members, except the sponsor, block.

The sponsor proceeds to update its share and compute the new group key; it can do this since it knows all necessary bkeys. Next, the sponsor broadcasts the new tree which contains all bkeys. All other members update their trees accordingly and compute the new group key It might appear wasteful to broadcast the entire tree to all members, since they already know most of the bkeys. However, since the sponsor needs to send a broadcast message to the group anyhow, it might as well include more information which is useful to the new member, thus saving one unicast message to the new member (which would have to contain the entire tree).

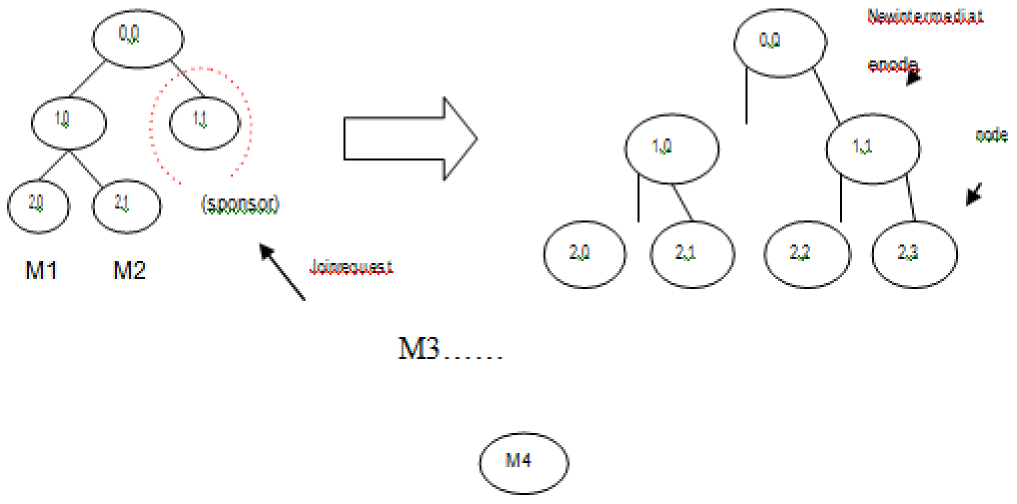

Fig3.4:M4 join the group

3.2.3 LEAVEOPERATION

we start with n users and assume that user Ud leaves the group. The sponsor in this case is the rightmost leaf node of the sub tree rooted at the leaving member's sibling node. each member updates its key tree by deleting the leaf node corresponding to Ud. The former sibling of Ud is promoted to replace Ud's parent node. The sponsor generates a new key share, computes all (S(l,v),bkey) pairs on the key-path up to the root, and broadcasts the new set of bkeys. This allows all members to compute the new groupkey.

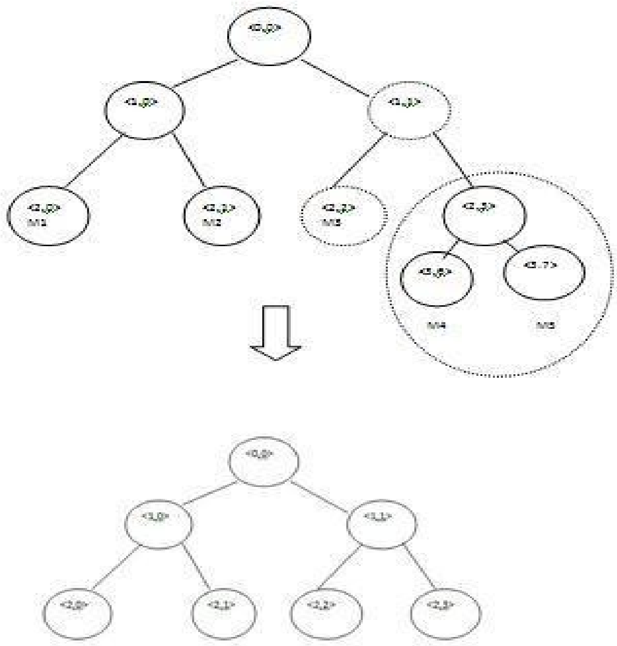

Fig3.5:m3 leave operation

Algorithm: Generalization of group Diffie Hellman Agreement algorithm

Initialization:

G be a group of order c, a be generator of G. The users are assumed to pick their secret logarithmic randomly from G_c.the mapping □: □→ □□is a bidirectional.

number of x children represents l_x. height of the tree is r.

S_x represents secret value generated by eachnode.

Phase1:

For all nodes z=x.i with l_x=0 (i)z selects a random $S_x \in Gc$ (ii) z→ □: □$^{□□}$

For all nodes z with $l_x \neq 0$

zselectsarandomu$_x$□□zwaitstoreceive□$^{□}$□.□forallk=1,………..,lx

.

3.3 REKEYING

A procedure in which a new cryptographic key using secure hash function(SHA) is generated in 256 bit key size of the (old) cryptographic key that it will replace. Contrast with Key update. Division keys are represented by the leave k-nodes in an key tree. Each user have a private key. for each division root employs an independent instance of the protocol to rekey the Corresponding division key when there is a change in membership happened in group.

- When a few members join at time t root simply broadcasts a new group key $GK^{(t+1)}$ encrypted under current group key $GK^{(t)}$ .Thus all members except the joining member can decrypt $GK^{(t+1)}$ from there key

Key generation time.message. for each of the joining members, root send it to the new group key $GK^{(t+1)}$ and its personal keys over a secure unicast channel.

➢ In leave rekeying, an exclusive key *Ki* can be used by root to exclude *i* (all users in *S*) in the sense that root can broadcast a new group key encrypted by *Ki* such that all users in *U* except *i* can decrypt the rekey message.

Algorithm: Rekeying

ot->ij:{K_1,...,k_{ij-1},...k_{ij+1},...,k_n},$GK^{(t+1)}$

After receiving the rekey notification message, every current member in *S*(*t*) can derive the new group key *GK*(*t*+1) by computing *GK*(*t*+1)=*h*(*GK*(*t*)).

IV. RESULTS AND DISCUSSION

The proposed has been implemented generalization group Diffie Hellman for Mobile Ad hoc Networks. Section 5.3 gives the group key generation results. Section 5.4 provides the sample result of member join rekey generation time. Section 5.5 provides the sample result of member leave re

Number of nodes	**Existing Method Group Key Key Generation time(Sec) time(inseconds)**	**Proposed Group Key Generation time(in Sec)**
16	0.15	0.13
18	0.41	0.28
20	0.71	0.61
46	1	0.9
68	1.28	0.95
92	1.52	1.13
100	2.75	2.23

The Table 4.1 list the group Key Generation time

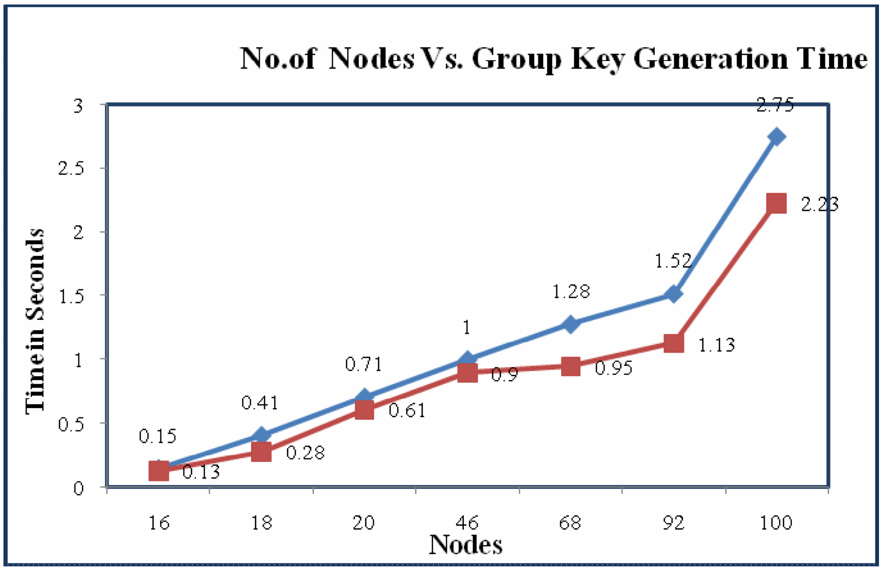

Fig4.1: Group Key Generation Time

4.1 USER JOINREKEYTIME

When a few users join in the tree the root node generate the group key and broadcast to all the group members. Inour proposed system reduce the rekey time.

Number of Nodes	Existing Method User Join time (in seconds)	Proposed Method User Join Rekey GenerationTime (in second)
16	0.15	0.14
18	0.4	0.28
20	0.7	0.72
46	0.99	0.96
68	1.1	1.03
92	1.48	1.7
100	2.52	2.36

Table 4.2 User Join RekeyTime

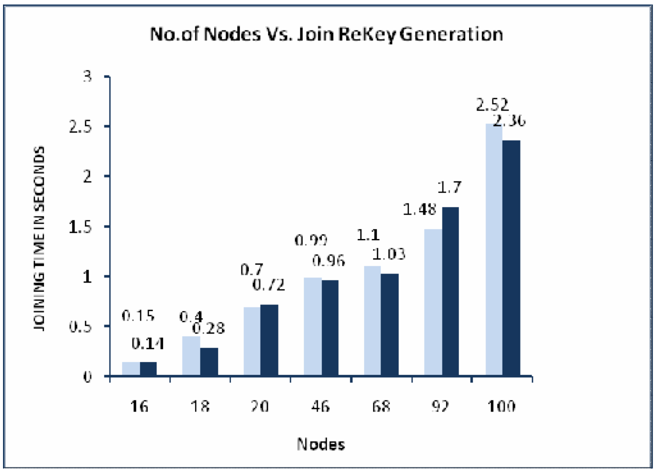

Fig4.2: Join Rekey Time

4.2 USER LEAVEREKEYTIME

In leave rekeying, an exclusive key *Ki* can be used by root to exclude *i*(all users in *S*) in the sense that root can broadcast a new group key encrypted by *Ki* such that all users in *U* except *i* can decrypt the rekey message. In generalize Diffie-Hellman reduce the rekey time whenever member leave. Compared with joinrekeying,
designing an efficient leave rekeying algorithm is much harder, because a revoked member always brings out information that may be used to encrypt future group keys. The user leave rekey time listed in table 5.7

No.of Nodes	Existing Method User leave time (in seconds)	Proposed Method Leave Rekey generation time (inseconds)
16	5.42	5.22
18	5.86	5.72
20	5.99	5.91
46	8	7.9
68	8.1	8
92	9.02	9
100	9.18	9.12

Table 4.3 user leave rekey time

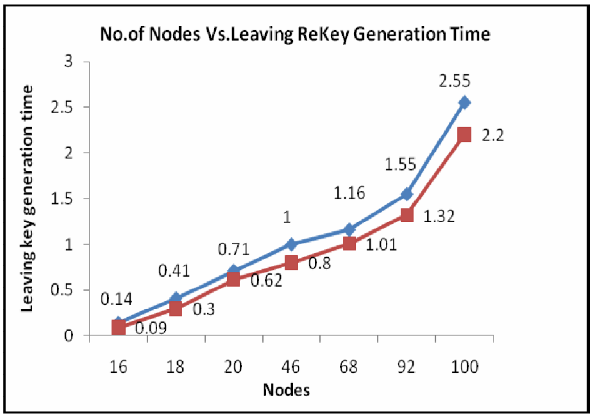

Fig4.3: Leaving Rekey Generation Time

V. CONCLUSION

In this proposed work, each node assigned a private key and public key. The root node is only needs to broadcast and compute the group key of joining user and leaving user. The new key value is computed by a one way function. When a user leaves, part of users can also compute the new keys by themselves. No matter when joining / leaving or switching the users in the group can reduce some rekeying time using hash algorithm. Our proposed scheme of the generalization group Diffie-Hellman key agreement algorithm is designed to avoid neighbor communication problem where the rekeying time, storage overhead can be dramatically reduced. It is expected that the proposed scheme can be practical solution for secure groupapplications.

REFERENCES

[1].Shanquan Jiang "Group Key Aggrement With Local Connectivity" IEEE Transacion on Dependable and Secure computing,vol.13,no.3 may/june 2016 pp:326-339.

[2].E.Bresson,O.Chevassut,and D. Pointcheval, "Dynamic group Diffie- Hellman key exchange under standard assumptions,"in Proc. 21th Int. Conf. Theory Appl. Cryptographic Techn., 2002, vol. 2332, pp. 321–336.

[3].M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system," in Proc. Adv. Cryptol., 1994, vol. 950,pp.275–286.

[4].K. Yongdae, P. Adrian, and G. Tsudik, "Group key agreement efficient in communication," IEEE Trans. Comput., vol. 53, no. 7,pp. 905–921, Jul. 2004.

[5].X. Lv, H. Li, and B. Wang, "Group key agreement for secure group communication in dynamic peer systems," J. Parallel Distrib.Comput., vol. 72, no. 10, pp. 1195–1200, 2012.

[6].Yang Yang1, 2, Yupu Hu2, Chunhui Sun2, Chao Lv2, Leyou Zhang3,"An Efficient Group Key Agreement Scheme for Mobile Ad-Hoc Networks," The International Arab Journal of Information Technology, Vol. 10, No. 1, January 2013. [7].E. Bresson, O. Chevassut, and D. Pointcheval, "Provably authenticated group Diffie-Hellman key exchange the dynamic case,"in Proc. 7th Int. Conf. Theory Appl. Cryptol. Inform. Security, 2001,vol. 2248, pp. 290–309.

[8].Lei Zhang a,b,⇑, QianhongWub,c, Bo Qin b,d, Josep Domingo-Ferrerb Provably secure one-round identity-based authenticated asymmetric group key agreement protocol L. Zhang et al. / Information Sciences 181 (2011)4318–4329.

[9].J. Katz and M. Yung, "Scalable protocols for authenticated group key exchange," in Proc. Adv. Cryptol., 2003, vol. 2729, pp. 110–125.

[10]. Wu, Y. Mu, W. Susilo, B. Qin, and J. Domingo-Ferrer,"Asymmetric group key agreement," in Proc. 28th Int. Conf. Theory Appl. Cryptographic Techn., 2009, vol. 5479, pp. 153–170.

[11]. Peter King Pong Au," Hierarchical Tree Approach to Group Key Management using the Group Diffie-Hellman Protocol," in University of British Columbia, 1999. [12]. M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman key distribution extended to group communication," in Proc. 3rd ACM Conf. Comput. Comm. Security, 1996, pp. 31–37.

[13].A.Joux,"Aoneroundprotocolfortripartite Diffie-Hellman,"inProc.4thInt. Symp. AlgorithmicNumberTheory,2000,pp.385–394.