A Novel Approach towards Cost Effective Region-Based Group Key Agreement Protocol for Ad Hoc Networks using Chinese Remainder Theorem

K. Kumar[1] , J.Nafeesa Begum[2] and Dr.V. Sumathy[3]
1. Research Scholar &Lecturer in CSE, Government College of Engg, Bargur- 635104, Tamil Nadu,India
pkk_kumar@yahoo.com
2. Research Scholar &Sr. Lecturer in CSE, Government College of Engg, Bargur- 635104, Tamil Nadu, India
nafeesa_jeddy@yahoo.com
3. Asst .Professor in ECE,Government College of Technology,Coimbatore, Tamil Nadu,India
sumi_gct2001@yahoo.co.in

Abstract *- A group key agreement (GKA) protocol is a mechanism to establish a cryptographic key for a group of participants, based on each one's contribution, over a public network. In ad-hoc networks, the movement of the nodes may quickly change the topology resulting in the increased overhead during messaging for topology maintenance. The Region-based schemes of ad-hoc networks, aim at handling topology maintenance, managing node movement and reducing overhead. In this paper, a simple, secure and efficient Region-based GKA protocol using CRTDH&TGDH well suited to dynamic ad-hoc networks is presented and also introduces a region-based contributory group key agreement that achieves the performance lower bound by utilizing a novel CRTDH and TGDH protocol called CRTDH&TGDH protocol. Both theoretical and experimental results show that the proposed scheme achieves communication, computation and memory cost is lower than the existing group key agreement schemes*.

Keywords: *Chinese Remainder theorem, Ad hoc networks, Region-Based key agreement*

1. Introduction

Ad hoc networks are networks composed of constrained devices communicating over wireless channels in the absence of any fixed infrastructure. Moreover, network composition is highly dynamic when devices leave /join the network quite frequently. Securing this type of networks become a more difficult task with additional challenges in the form of lack of trusted third parties, expensive communication, ease of interception of messages and limited computational capabilities of the devices. In ad hoc networks, the key distribution techniques are not useful as there is not enough trust in the network so as to agree on a key decided by one member or some central authority.

Group Key Agreement (GKA) protocols [4,5, 6, 7, &8], which enable the participants to agree on a common secret value based on each participant's contribution, seem to provide a good solution. Also when group composition changes, group controller can employ supplementary key agreement protocols to get a new group key.

Secure group communication (SGC) is the process by which members in the group can securely communicate with each other and information being shared is inaccessible to anybody outside the group. A SGC protocol should efficiently manage the group key when the members join and leave the groups. This is considerably done in MANETs with its high mobility and dynamic network topology.

SGC should satisfy the following properties: shared group key, backward secrecy, forward secrecy, multiple users to join/leave simultaneously, efficiency with minimum amount of computation and communication.

A number of protocols have been proposed to handle SGC over MANETs and none of the above properties are satisfied. In our paper, we have proposed the important MANET features with respect to SGC: No pre-shared secret since the participating members are not known before hand, No centralized Trusted Authority (TA), Optimized battery power, all nodes have balanced load, position and capability and Nodes are highly mobile.

Earlier we have proposed [1] a Contributory Group Key Agreement which fulfills the efficacious lower bound by utilizing a novel GECDH & TGECDH in which a leader communicates with the member in the same region using a regional key KR. The Outer Group Key is derived from subgroup leaders. In this approach, a GECDH protocol needs member serialization among the members. It is not a desirable property of ad hoc networks.

We propose a communication and computation efficient group key agreement protocol in ad-hoc network. In large and high mobility ad hoc network s, it is not possible to use a single group key for the entire network because of the enormous cost of computation and communication in rekeying. So, we divide the group into several subgroups; let each subgroup has its subgroup key shared by all members of the subgroup. Each group has sub group controller node and gateway node, in which the sub group controller is controller of each subgroup and gateway node is controller of among subgroups. Let each gateway member contribute a partial key to agree with a common Outer group key among the subgroups.

In this paper we have proposed a Region-based Contributory GKA that achieves the performance lower bound by utilizing a novel Chinese Remainder Theorem Diffie-Hellman (CRTDH) and TGDH protocol for secure group communication over MANETs. In this approach, the member serialization between the subgroup members is eliminated.

The contribution of this work includes:

1. In this paper, we propose a new efficient method for solving the group key management problem in ad-hoc network. This protocol provides efficient, scalable and reliable key agreement service and is well adaptive to the mobile environment of ad-hoc network.

2. We introduce the idea of subgroup and subgroup key and we uniquely link all the subgroups into a tree structure to form an outer group and outer group key. This design eliminates the centralized key server. Instead, all hosts work in a peer-to-peer fashion to agree on a group key. We use Region-Based Group Key Agreement (RBGKA) as the name of our protocol. Here we propose a region based group key agreement protocol for ad hoc networks using Chinese Remainder Theorem called Region-Based CRTDH & TCDH protocol.

3. We design and implement Region-Based Group key agreement protocol using Java and conduct extensive experiments and theoretical analysis to evaluate the performance like memory cost, communication cost and computation cost of our protocol for Ad- Hoc network.

The rest of the paper is, Section 2 describes the Chinese Remainder Theorem .Section 3 presents the proposed CRTDH&TGDH schemes. Section 4 describes CRTDH Key Agreement Protocol. Section 5 describes the TGDH Protocol. Section 6 describes the experimental results. Section 7 describes the complexity analysis and finally Section 8 concludes the paper.

2. Chinese Remainder Theorem(CRT)

The Chinese Remainder Theorem is used as the theoretical foundation in many cryptosystems.

Suppose $N_1,\ldots,N_r$ are pairwise relatively prime positive integer, i.e., $\gcd(N_i,N_j) = 1$ if $i \neq j$, and let $N= N_1.N_2\ldots N_r$. For any given integers $a_1,\ldots,a_r$, consider the following system of congruences:

$$X \equiv a_1 \bmod N_1$$
$$X \equiv a_2 \bmod N_2$$
$$\vdots$$
$$X \equiv a_r \bmod N_r$$

Then the system has a unique solution modulo N, namely:

$$x \equiv \sum_{i=1}^{r} a_i n_i y_i \bmod N$$

Where $n_i = N/N_i$ and $y_i = n_i^{-1} \bmod N_i$ (i.e. y_i is the multiplicative inverse of n_i modulo N_i).

3. Proposed Scheme

3.1. Motivation

There has been a growing demand in the past few years for security in collaborative environments deployed for emergency services where our approach can be carried out very efficiently are shown in Fig.1.Confidentiality becomes one of the top concerns to protect group communication data against passive and active adversaries. To satisfy this requirement, a common and efficient solution is to deploy a group key shared by all group application participants. Whenever a member leaves or joins the group, or whenever a node failure or restoration occurs, the group key should be updated to provide forward and backward secrecy.

Figure.1. Secure Group Applications

Therefore, a key management protocol that computes the group key and forwards the rekeying messages to all legitimate group members is central to the security of the group application.

In many secure group applications [1, 3], a Region based contributory GKA schemes may be required. In such cases, the group key management should be both efficient and fault-tolerant. In this paper,

Figure.2. Battlefield Scenario

we describe a military scenario (Figure.2). A collection of wireless mobile devices are carried by soldiers or a Battlefield tanks. These mobile devices cooperate in relaying packets to dynamically establish routes among themselves to form their own network "on the fly". However, all nodes except the one with the tank, have limited battery power and processing capacities. For the sake of power- consumption and computational efficiency, the tank can work as the Gateway member while a contributed group key management scheme is deployed.

3.2. System Model

3.2.1. Overview of Region-Based Group Key Agreement Protocol:

The goal of this paper is to propose a communication and computation efficient group key establishment protocol in ad-hoc network. The idea is to divide the multicast group into several subgroups, let each subgroup has its subgroup key shared by all members of the subgroup. Each Subgroup has subgroup controller node and a Gateway node, in which Subgroup controller is controller of subgroup and a Gateway node is controller of subgroups controller.

For example, in Figure.3, all member nodes are divided into number of subgroups and all subgroups are linked in a tree structure as shown in Figure.4.

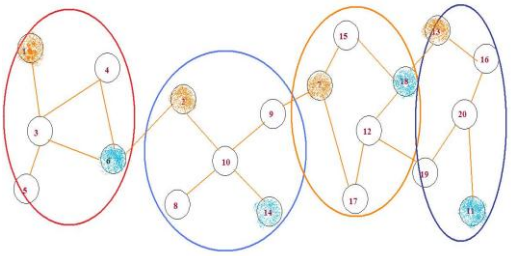

Figure.3: Members of group are divided into subgroups

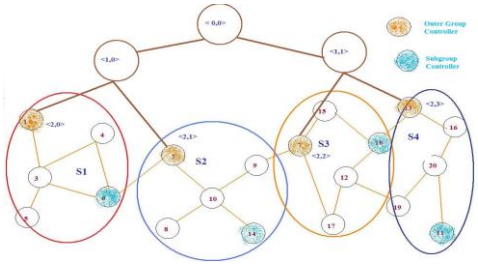

Figure.4: Subgroups link in a Tree Structure

The layout of the network is as shown in below figure.5.

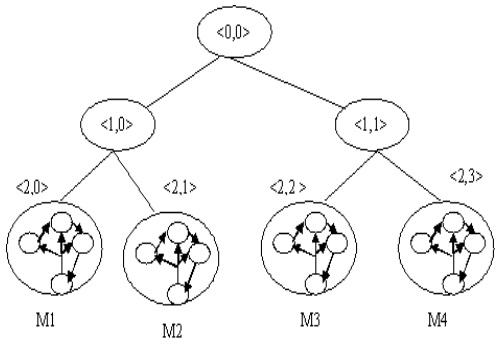

Figure.5. Region based Group Key Agreement

One of the members in the subgroup is subgroup controller. The last member joining the group acts as a subgroup controller. Each outer group is headed by the outer group controller. In each group, the member with high processing power, memory, and Battery power acts as a gateway member. Outer Group messages are broadcast through the outer group and secured by the outer group key while subgroup messages are broadcast within the subgroup and secured by subgroup key.

Let N be the total number of group members, and M be the number of the subgroups in each subgroup, then there will be N/M subgroups, assuming that each subgroup has the same number of members.
There are two shared keys in the Region-Based Group Key Agreement Scheme:

1. Outer Group Key (KG)is used to encrypt and decrypt the messages broadcast among the subgroup controllers.
2. The Subgroup Key (KR) is used to encrypt and decrypt the Sub Group level messages broadcast to all sub group members.

In our Region-Based Key Agreement protocol shown in Fig.5 a Subgroup Controller communicates with the member in the same region using a Regional key (i.e Sub group key) KR. The Outer Group key KG is derived from the Outer Group Controller. The Outer Group Key KG is used for secure data communication

among subgroup members. These two keys are rekeyed for secure group communications depending on events that occur in the system.

Assume that there are total N members in Secure Group Communication. After sub grouping process (Algorithm 1), there are S subgroups M_1, M_2… M_s with n_1, n_2 …n_s members.

Algorithm. 1. Region-Based Key Agreement protocol

1. The Subgroup Formation
   The number of members in each subgroup is
   N / S < 100.

Where, N – is the group size. And S – is the number of subgroups.

Assuming that each subgroup has the same number of members.

2. The Contributory Key Agreement protocol is implemented among the group members. It consists of three stages.
   a. To find the Subgroup Controller for each subgroup.
   b. CRTDH protocol is used to generate one common key for each subgroup headed by the subgroup controller.
   c. Each subgroup gateway member contributes partial keys to generate a one common backbone key (i.e Outer group Key (KG)) headed by the Outer Group Controller using TGDH protocol.
3. Each Group Controller (sub /Outer) distributes the computed public key to all its members. Each member performs rekeying to get the respected group key.

A Regional key KR is used for communication between a subgroup controller and the members in the same region. The Regional key KR is rekeyed whenever there is a membership change event and subgroup joins / leaves and member failure. The Outer Group key KG is rekeyed whenever there is a join / leave subgroup controllers and member failure to preserve secrecy.

The members within a subgroup use Chinese Remainder theorem Group Diffie-Hellman Contributory Key Agreement (CRTDH). Each member within a subgroup contributes his share in arriving at the subgroup key. Whenever membership changes occur, the subgroup controller or previous member initiates the rekeying operation.

The gateway member initiates communication with the neighbouring member belonging to another subgroup and mutually agree on a key using Tree-Based Group Diffie-Hellman contributory Key Agreement(TGDH) protocol to be used for inter subgroup communication between the two subgroups. Any member belonging to one subgroup can communicate with any other member in another subgroup through this member as the intermediary. In this way adjacent subgroups agree on outer group key. Whenever membership changes occur, the outer group controller or previous group controller initiates the rekeying operation.

Here, we prefer the subgroup key to be different from the key for backbone. This difference adds more freedom of managing the dynamic group membership. In addition using this approach can potentially save the communication and computational cost.

3.3. Network Dynamics

The network is dynamic in nature. Many members may join or leave the group. In such case, a group key management system should ensure that backward and forward secrecy is preserved.

3.3.1. Member Join

When a new member joins, it initiates communication with the subgroup controller. After initialization, the subgroup controller changes its contribution and sends public key to this new member. The new member receives the public key and acts as a group controller by initiating the rekeying operations for generating a new key for the subgroup. The rekeying operation is as follows.

$$\text{New node} \xrightarrow{\text{Join request}} \text{Subgroup Controller}$$

$$\text{Subgroup Controller} \xrightarrow{\text{change its contribution and send public key to}} \text{New Node}$$

$$\text{New Node} \xrightarrow{\text{Act as}} \text{New Subgroup Controller}$$

$$\text{New Subgroup Controller} \xrightarrow{\text{puts its contribution to all the public key value \& Multicast this public key value to}} \text{the entire member in the subgroup}$$

$$\text{Each Member} \xrightarrow{\text{put is contribution to the public value \& Compute}} \text{New Subgroup Key}$$

3.3.2. Member Leave:

3.3.2.1. When a Subgroup member leaves

When a member leaves the Subgroup Key of the subgroup to which it belongs must be changed to preserve the forward secrecy. The leaving member informs the subgroup controller. The subgroup controller changes its private key value, computes the public value and broadcasts the public value to all the remaining members. Each member performs rekeying by putting its contribution to public value and computes

the new Subgroup Key. The rekeying operation is as follows.

$$\text{Leaving Node} \xrightarrow{\text{Leaving Message}} \text{Subgroup Controller}$$

$$\text{Subgroup Controller} \xrightarrow[\text{changes its private key value, compute the public key value and Multicast the public key value to}]{} \text{All the remaining Member}$$

$$\text{Each Member} \xrightarrow{\text{Perform Rekeying and Compute}} \text{New Subgroup Key}$$

3.3.2.2.When Subgroup Controller Leaves:

When the Subgroup Controller leaves, the Subgroup key used for communication among the subgroup controllers need to be changed. This Subgroup Controller informs the previous Subgroup Controller about its desire to leave the subgroup which initiates the rekeying procedure. The previous subgroup controller now acts as a Subgroup controller. This Subgroup controller change its private contribution value and computes all the public key value and broadcasts to all the remaining member of the group. All subgroup members perform the rekeying operation and compute the new subgroup key. The rekeying operation is as follows.

$$\text{Leaving Subgroup Controller} \xrightarrow{\text{Leaving Message}} \text{Old Subgroup Controller}$$

$$\text{Old Subgroup Controller} \xrightarrow[\text{change its private value,compute the all public key value and Multicast}]{} \text{Remaining Member in the group}$$

$$\text{Subgroup Member} \xrightarrow{\text{Perform Rekeying and Compute}} \text{New Subgroup Key}$$

3.3.2.3. When Outer Group Controller Leaves:

When a Outer group Controller leaves, the Outer group key used for communication among the Outer groups need to be changed. This Outer group Controller informs the previous Outer group Controller about its desire to leave the Outer group which initiates the rekeying procedure. The previous Outer Group controller now becomes the New Outer group controller. This Outer group controller changes its private contribution value and computes the public key value and broadcast to the entire remaining member in the group. All Outer group members perform the rekeying operation and compute the new Outer group key. The rekeying operation is as follows.

$$\text{Leaving Outer group Controller} \xrightarrow{\text{Leaving Message}} \text{Old Outer group Controller}$$

$$\text{Old Outer group Controller} \xrightarrow[\text{change its private value,compute the all public key value and Multicast}]{} \text{Remaing Member in the Outer group}$$

$$\text{Outer group Member} \xrightarrow{\text{Perform Rekeying and Compute}} \text{New Outer group Key}$$

3.3.2.4. When Gateway member leaves

When a gateway member leaves the subgroup, it delegates the role of the gateway to the adjacent member having high processing power, memory, and Battery power and acts as a new gateway member. Whenever the gateway member leaves, all the two keys should be changed. These are

i. Outer group key among the subgroup.
ii. Subgroup key within the subgroup.

In this case, the subgroup controller and outer group controller perform the rekeying operation. Both the Controller leave the member and a new gateway member is selected in the subgroup, performs rekeying in the subgroup. After that, it joins in the outer group. The procedure is same as joining the member in the outer group.

3.4. Communication Protocol:

The members within the subgroup have communication using subgroup key. The communication among the subgroup members takes place through the gateway member.

3.4.1. Communication within the Subgroup:

The sender member encrypts the message with the subgroup key (KR) and multicasts it to all member in the subgroup. The subgroup members receive the encrypted message, perform the decryption using the subgroup key (KR) and get the original message. The communication operation is as follows.

$$\text{Source Member} \xrightarrow{E_{KR}[\text{Message}] \text{ \& Multicast}} \text{Destination Member}$$

$$\text{Destination Member} \xrightarrow{D_{KR}\left[E_{KR}[\text{Message}]\right]} \text{Original Message}$$

3.4.2. Communication among the Subgroup:

The sender member encrypts the message with the subgroup key (KR) and multicasts it to all members in the subgroup. One of the members in the subgroup acts as a gate way member. This gateway member decrypts the message with subgroup key and encrypts with the outer group key (KG) and multicasts to the entire gateway member among the subgroup. The destination gateway member first decrypts the message with outer group key. Then encrypts with subgroup key and multicasts it to all members in the subgroup. Each member in the subgroup receives the encrypted message and performs the decrypts operation using subgroup key and gets the original message. In this way the region-based group key agreement protocol performs the communication. The communication operation is as follows.

$$\text{Source Member} \xrightarrow{E_{KR}[\text{Message}] \text{ \& Multicast}} \text{Gateway Member}$$

$$\text{Gateway Member} \xrightarrow{D_{KR}\left[E_{KR}[\text{Message}]\right]} \text{Original Message}$$

$$\text{Gateway Member} \xrightarrow{E_{KG}[\text{Message}] \text{ \& Multicast}} \text{Gateway Member [Among Subgroup]}$$

$$\text{Gateway Member} \xrightarrow{D_{KG}\left[E_{KG}[\text{Message}]\right]} \text{Original Message}$$

$$\text{Gateway Member} \xrightarrow{E_{KR}[\text{Message}] \text{ \& Multicast}} \text{Destination Member}$$

$$\text{Destination Member} \xrightarrow{D_{KR}\left[E_{KR}[\text{Message}]\right]} \text{Original Message}$$

4. CRTDH Key Agreement Protocol [2, 3]

4.1.Group Key Establishment

Every group member U_i does as follows.

Step 1.1. Selects a DH private share *xi*, computes

$$y_i = g^{xi} \bmod p.$$

Step 1.2. Broadcast y_i.

Step 1.3. Computes

$$m_{ij} = \begin{cases} y_j^{x_i} \bmod p & \text{if } y_j^{x_i} \text{ mo} \\ p - y_j^{x_i} \bmod p & \text{otherwise} \end{cases}$$

Step 1.4. For a given $j \neq i$, U_i chooses P_{ij} such that $\gcd(P_{ij}, m_{ij}) = 1$

Step 1.5. For $j \neq i$, U_i uses the following new congruences to replace the original ones:

$$crt_{ij} \equiv k_i \pmod{m_{ij}}$$
$$crt_{ij} \equiv D_i \pmod{P_{ij}}$$

where K_i & D_i *are random.* U_i also broadcasts the set of pairs

$$crt_i = \{(U_j, crt_{ij}) : j \neq i\},$$

Each group member U_j finds his own matched crt_{ij} ($i \neq j$) from crt_i broadcast by U_i.

Step 1.6. U_i Computes $\boldsymbol{crt_{ji} \bmod m_{ij}}$, $j \neq i$,which must be k_j since $m_{ij} = m_{ji}$, and then computes $\boldsymbol{GK = k_1 \oplus k_2 \oplus \ldots \oplus k_n}$, which is the group key.

The illustration of three users CRDH Key establishment is shown in Figure.7 below and The implementation as shown in Figures.16.

USER - I	USER - II	USER - III
G =5 , p = 32713	G =5 , p = 32713	G =5 , p = 32713
Step 1.1:	**Step 1.1:**	**Step 1.1:**
X_1 = 81234	X_2 = 96727	X_3 = 52410
Y_1 = 9506	Y_2 = 25139	Y_3 = 21122
Step 1.2 .	**Step 1.2 .**	**Step 1.2 .**
Y_1 = 9506	Y_1 = 9506	Y_1 = 9506
Y_2 = 25139	Y_2 = 25139	Y_2 = 25139
Y_3 = 21122	Y_3 = 21122	Y_3 = 21122
Step 1.3:	**Step 1.3:**	**Step 1.3:**
M_{12} = 32221	M_{21} = 32221	M_{31} = 20490
M_{13} = 20490	M_{23} = 21291	M_{32} = 21291
Step 1.4 :	**Step 1.4 :**	**Step 1.4 :**
K_1 =5876	K_2 =13390	K_3 =4387
D_1=5007	D_2=11559	D_3=3266
P_{12} =16935	P_{21} =3131	P_{31} =2107
P_{13} =16769	P_{23} =5161	P_{32} =15634
Step 1.6 :	**Step 1.6 :**	**Step 1.6 :**
Crt_{12} =296016521377	Crt_{21} =1329951302826	Crt_{31} =1557704142 37
Crt_{13} =1823983436126	Crt_{23} =1377419363491	Crt_{32} =1149926807932
$crt_1^i = \{(User_1, 2960165218377), (User_3, 1823983436126)\}$	$crt_2^i = \{(User_1, 1329951302826), (User_3, 1377419363491)\}$	$crt_3^i = \{(User_1, 155770414237), (User_2, 1149926807932)\}$
Step 1.6 :	**Step 1.6 :**	**Step 1.6 :**
K_1 = 5876	K_1 = 5876	K_1 = 5876
K_2 = 13390	K_2 = 13390	K_2 = 13390
K_3 = 4387	K_3 = 4387	K_3 = 4387
Group Key	**Group Key**	**Group Key**
$GK = k_1 \oplus k_2 \oplus k_3$	$GK = k_1 \oplus k_2 \oplus k_3$	$GK = k_1 \oplus k_2 \oplus k_3$
GK = 5876⊕13390⊕3387	GK = 5876⊕13390⊕3387	GK = 5876⊕13390⊕3387
GK =13209	GK =13209	GK =13209

Fig: 7. CRTDH Key Establishment for Three Users

4.2.The CRTDH join operation

Suppose that U_{n+1} joining the group is shown in Fig.8 below & The implementation is shown in Fig.17.

Step 2.1. U_i ($1 \leq i \leq n$) compute the hash value $h(GK)$. One of them, say U_1, transmits $h(GK)$ and y_i ($1 \leq i \leq n$) to U_{n+1}.

Step 2.2. U_{n+1} executes Steps 1.1-1.2, executes Step 1.3 only for $m_{n+1,t}$ ($1 \leq t \leq n+1$; $t \neq$ n+1), and executes Steps 1.4-1.5 only for $crt_{n+1,t}$ ($1 \leq t \leq n+1$; $t \neq$ n+1), i.e., compute and broadcast y_{n+1} and crt_{n+1}.

Step 2.3. U_i ($1 \leq i \leq n+1$) recovers k_{n+1} *by* using the method in Step 1.6, and computes the new group key as $\boldsymbol{GK_{new} = h(GK) \oplus k_{n+1}}$.

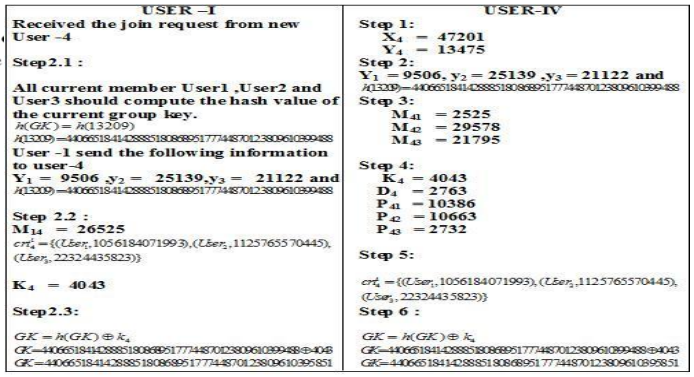

Figure: 8. User 4 Join the Group

4.3.The CRTDH leave operation

Suppose that $n > s > 1$ & *Us is* going to leave given in Fig.9 below & The implementation is shown in Fig.18.

Step 3.1. One of the U_i, say U_1, repeats Steps 1.4 -1.5 *with* a newk_1, broadcasts the new $crt_1 = \{crt_1, t : 2 \leq t \leq n\}$,& computes the new group key $\boldsymbol{GK_{new} = GK \oplus k_1}$.

Step 3.2. U_i ($2 \leq i \leq n$) recovers k_1 from crt_1 and then compute the new group key $\boldsymbol{GK_{new} = GK \oplus k_1}$.

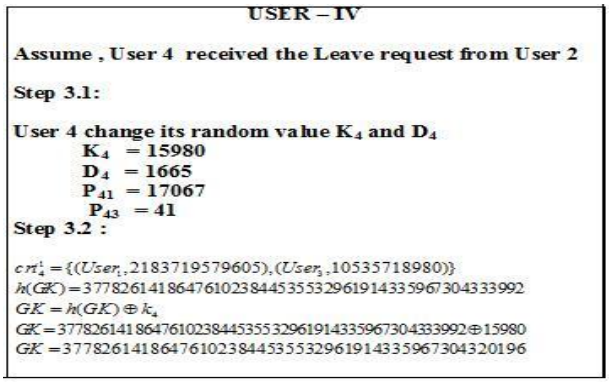

Figure.9. After User 2 leave the group

5. TGDH Protocol [4, 5, 6 & 7]

One of the main features of our work is the use of key trees in fully distributed contributory key agreement. Figure.10 shows an example of a key tree. The root is located at level 0 and the lowest leaves are at level *h*. Since we use binary trees, every node is either a leaf or a parent of two nodes. The nodes are denoted $<l, v>$, where $0 \leq v \leq 2^l - 1$ since each level *l*

hosts at most 2^l nodes. Each node $<l , v>$ is associated with the key $K_{<l ,v>}$ and the *blinded key* (bkey) $BK_{<l ,v>} = f(K_{<l ,v>})$, where the function f (.) is modular exponentiation in prime order groups, that is, $f(k) = \alpha^k \bmod p$ (equivalent to the Diffie–Hellman protocol.

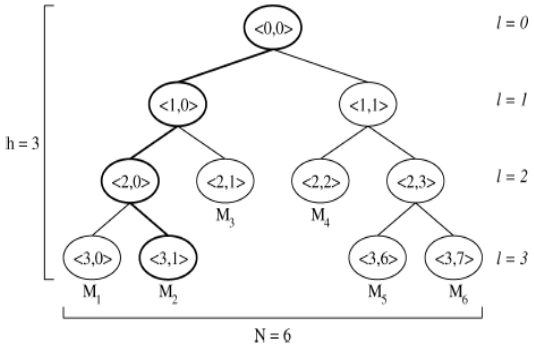

Figure 10. A Key Tree.

5.1 Join Protocol

Assume the group has n members: $\{M_1, \ldots, M_n\}$. The new member M_{n+1} initiates the protocol by broadcasting a join request message that contains its own bkey $BK_{<0,0>}$. Each current group controller 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 group controller is the rightmost leaf in the sub tree rooted at the insertion node. The group controller proceeds to update its share and passed all bkeys tree structure to new joining member. The new joining member acts as the group controller and computes the new group key. Next, the group controller broadcasts the new tree that contains all bkeys. All other members update their trees accordingly and compute the new group key.

5.1.1 . Illustrating with examples

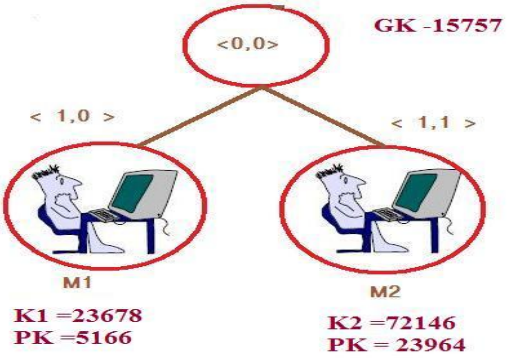

Figure: 11. User M1&M2 Join using TGDH

User M_1 and User M_2 are going to exchange their keys: Take **g = 5** and **p = 32713**. User M_1's private key is **23678**, so M_1's public key is **5166**. User M_2's private key is **72146**, so M_2's public key is **239640**. The Group key is computed (Figure.11) as User M_1 sends its public key **5166** to user M_2, the User M_2 computes their group key as **15757**. User M_2 sends its public key **23964** to user M_1, then the user M_1 computes their group key as **15757**. Here, Group controller is User M_2.

When 3rd node Joins

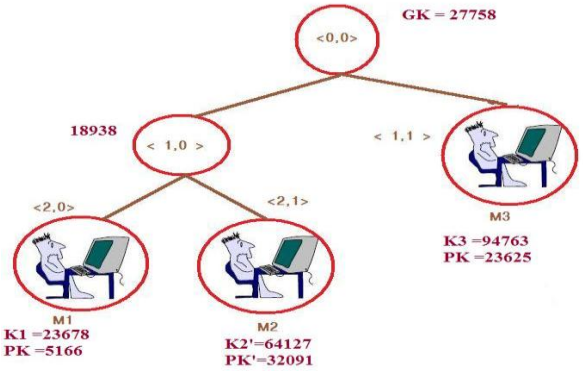

Figure 12. User M_3 Join the Group

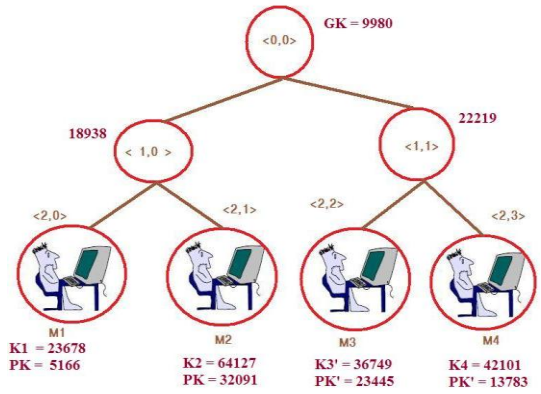

Figure. 13. User M_4 Join the group

When User joins the group, the old group controller M_2 changes its private key value from **72146** to **64127** and passes the public key value and tree to User M_3. Now, M_3 becomes group controller. Then, M_3 generates the public key **23625** from its private key as **94763** and computes the group key as **27758** shown in Figure.12. M_3 sends Tree and public key to all users. Now, user M_1 and M_2 compute their group key. The same procedure followed by joining the User 4 as shown in Fig. 13. The implementations are as shown in Figures.19, 20&21.

5.2. Leave Protocol

There are two types of leave, 1.Ordinary member leave and 2.Group controller leave

5.2.1. Ordinary member leave

When user M_2 leaves (Figure.14) the group, then the Group controller changes its private key **42101** to **27584** and group key is recalculated as **7914**. After that, it broadcast its Tree and public key value to all users in the group. Then, the new group key will be generated by the remaining users. The implementation is as shown in Fig.22.

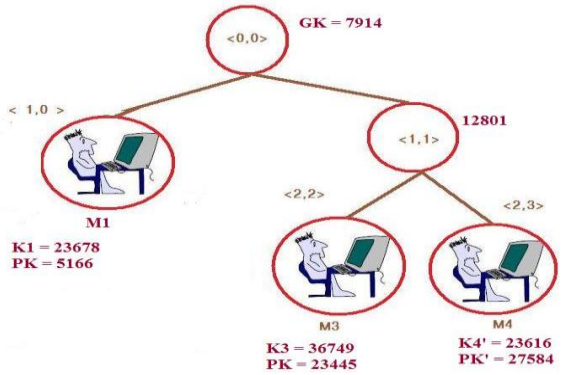

Figure.14. User M_2 Leave from the Group

5.2.2. When a Group controller leaves

When a Group controller leaves (Figure.15) from the group, then its sibling changes its private key value 36749 to **14214** and recalculates the group key as **6576**. After that, the same steps are followed by ordinary member leave method. The implementation is as shown in Figure.23.

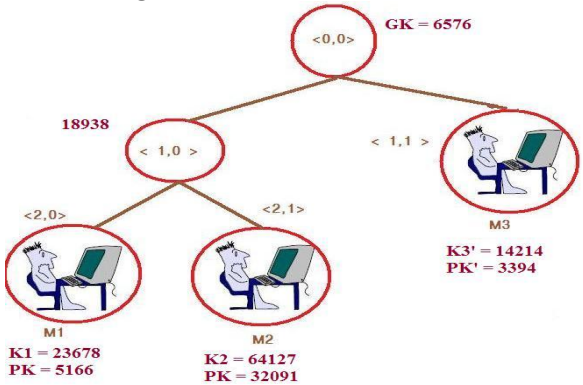

Figure.15. Group Controller Leave from the Group

6. Experimental Results and Discussion

The experiments were conducted on sixteen Laptops running on a 2.4 GHz Pentium CPU with 2GB of memory and 802.11 b/g 108 Mbps Super G PCI wireless cards with Atheros chipset. To test this project in a more realistic environment, the implementation is done by using Net beans IDE 6.1, in an ad-hoc network where users can securely share their data. This project integrates with a peer-to-peer (P2P) communication module that is able to communicate and share their messages with other users in the network which is described below.

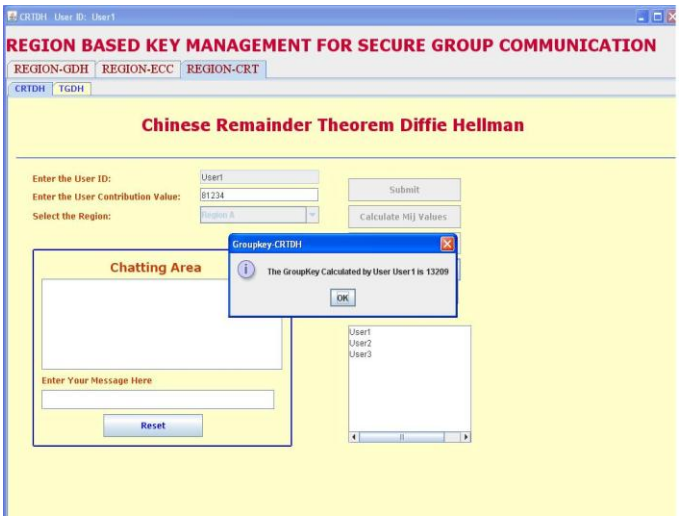

Figure.16. Group Key of User 1, 2 &3

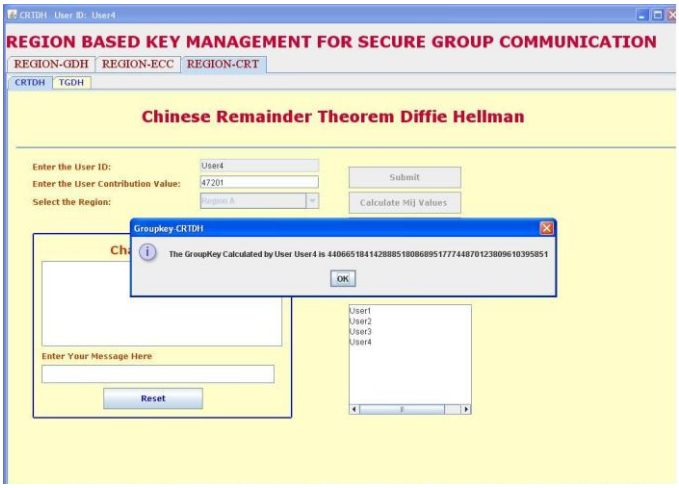

Figure.17. Group Key after User4 Join.

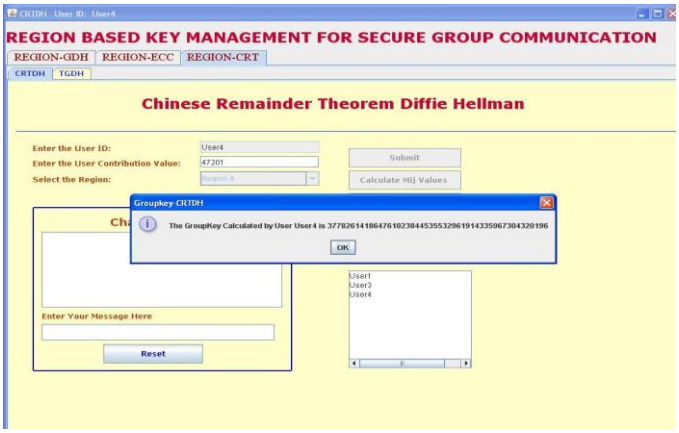

Figure.18. Group Key after User2 Leave

Figure19. Group Key of User M_1&M_2

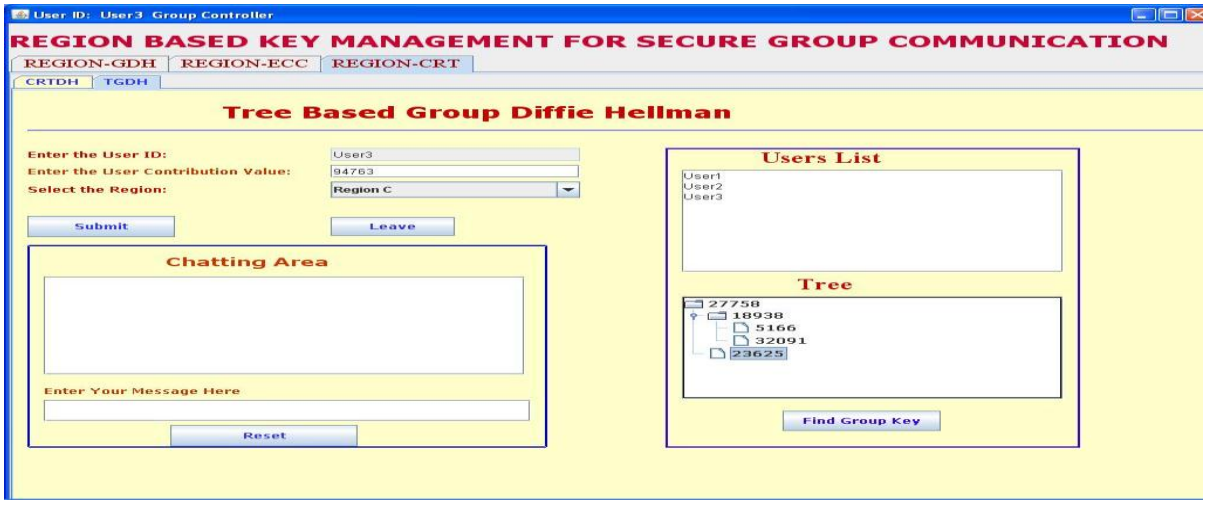

Figure 20. Group Key of User M_1,M_2&$M_3$$M_1$&$M_2$

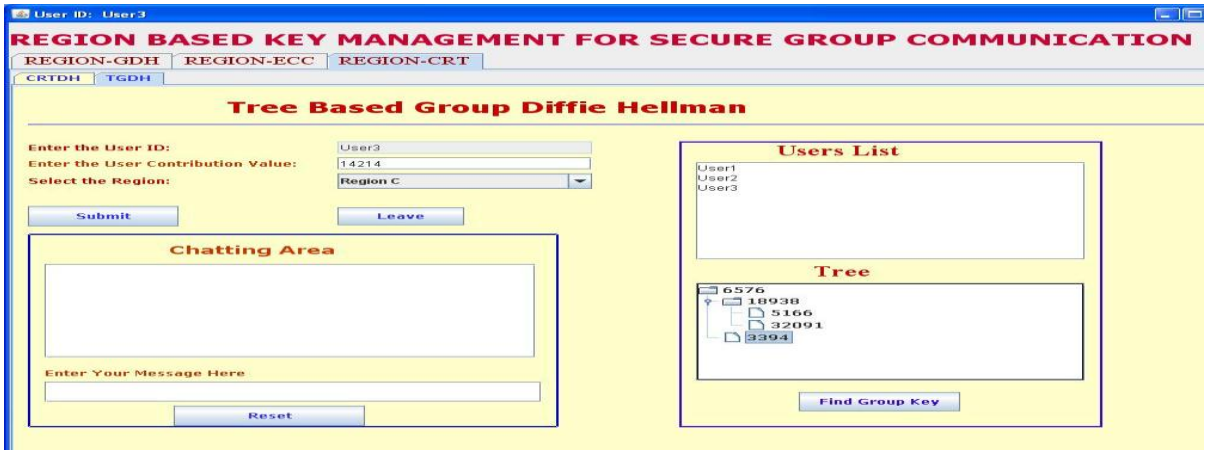

Figure 21. Group Key of User M_1, M2, M3 & M_4

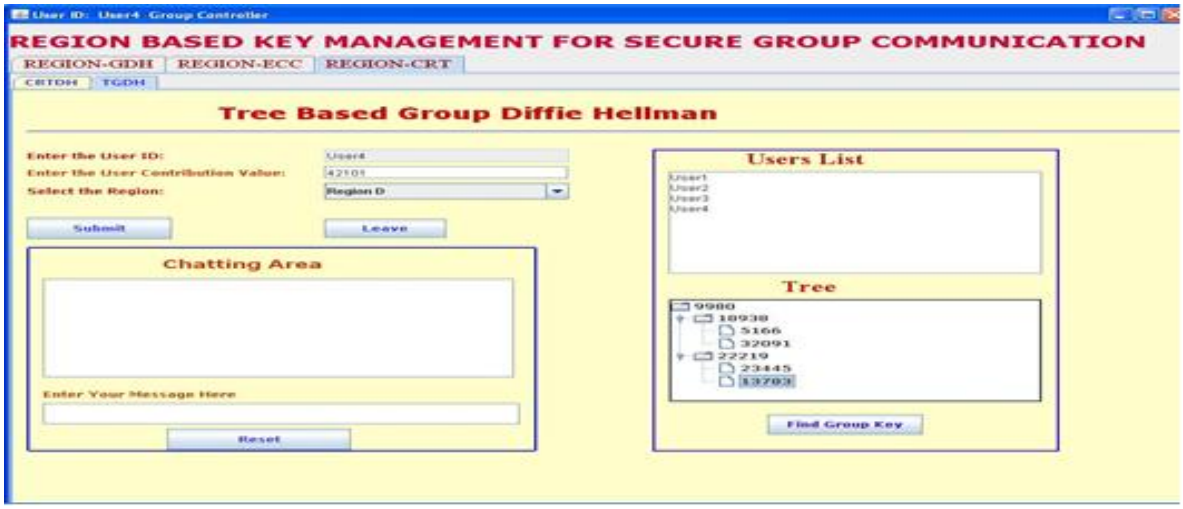

Figure. 22. Group Key after M_2 Leave

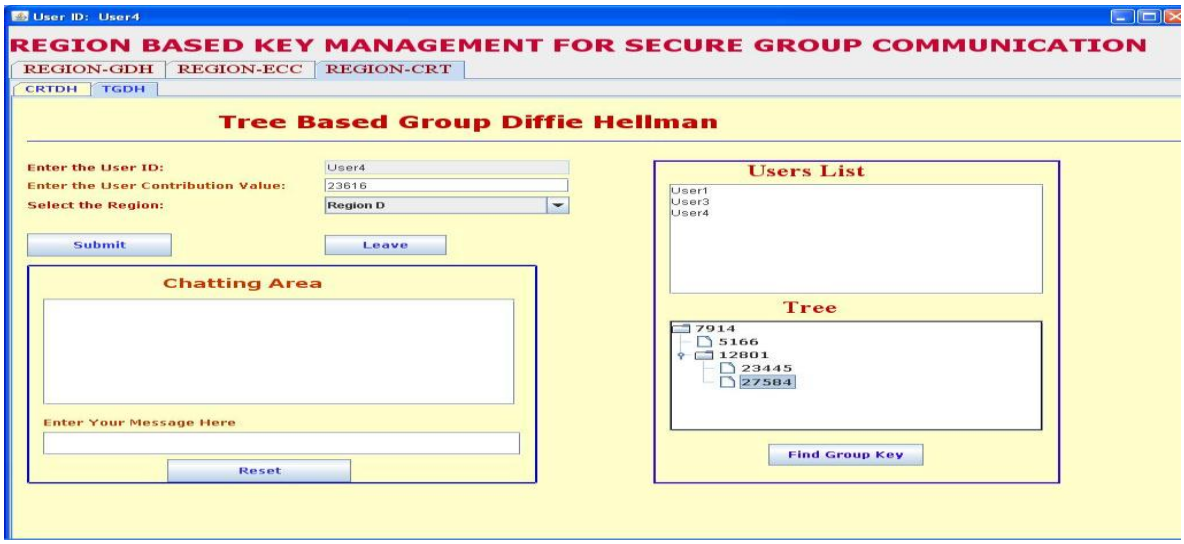

Figure.23. Group Key after Group Controller Leave

7. Complexity Analysis

7.1. Memory Costs:

Our approach consumes very less memory compared to TGDH and CRTDH when members go on increasing.

7.2. Communication Costs:

Our approach consumes less bandwidth when compare to CRTDH and TGDH. TGDH depends on trees height, balance of key tree, location of joining tree, and leaving nodes. But this approach depends on the number of members in the subgroup, number of Group Controller, and height of tree.

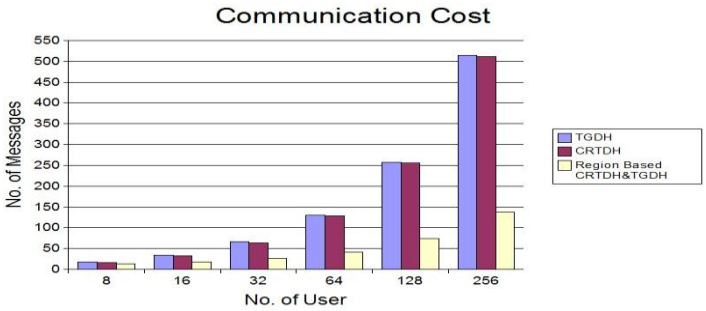

Figure.24. Communication Cost

Considering (Figure.24) there are 256 members in a group our approach consumes only 29% of Bandwidth when compare to CRTDH and TGDH.

7.3. Computation Costs:

The Computational costs depend on the Number of exponentiations. CRTDH has high computation costs as it depends on the number of members and group size respectively. The cost increases as the members and group size increases. But our approach spends a little on this computation.

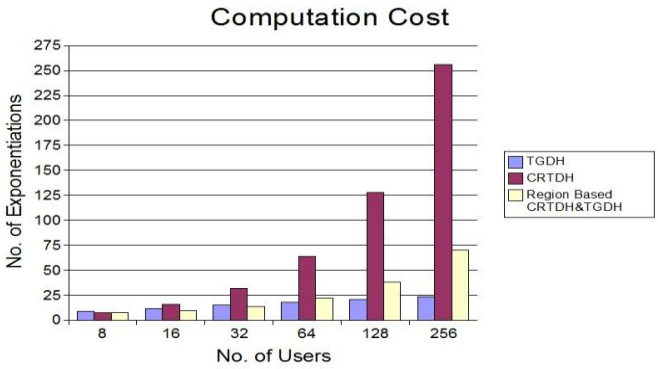

Figure.25. Computation Cost

Consider (Figure.25) there are 256 members in a group. Our approach consumes nearly 27% of less exponentiation when compared to CRTDH. But in this case TGDH is less compared to our approach. But this occurs because of additive nature of CRTDH and TGDH. Performance wise our approach leads the other two methods to overcome member serialization, even for the very large groups.

8. Conclusion

In this paper, a region-based key agreement scheme has been proposed and implemented, which can enhance the secure group communication performance by using multiple group keys. In contrast to other existing schemes using only single key, the new proposed scheme exploits asymmetric key, i.e an Outer group Key and multiple Subgroup keys, which are generated from the proposed Region-Based key agreement algorithm. By using a set comprising an outer group key and subgroup keys a region-based scheme can be efficiently distributed for multiple secure groups. Therefore, the number of rekeying messages, computation and memory can be dramatically reduced. Compared with other schemes, the new proposed Region-Based scheme can significantly reduce the storage and communication overheads in the rekeying process, with acceptable computational overhead. It is expected that the proposed scheme can be the practical solution for secure group applications, especially for Battlefield Scenario.

References

[1]. Kumar. K , Sumathy. V and J. Nafeesa Begum, "Efficient Region-Based Group Key Agreement protocol for Ad Hoc networks using Elliptic Curve Cryptography", IEEE International Advance computing Conference,6-7 March 2009,pp.1052-1060.

[2]. Spyros Magliveras, Wandi Wei and Xukai Zou," Notes on the CRTDH Group Key Agreement Protocol", The 28th International Conference on Distributed Computing Systems Workshops,2008,pp-406-411.

[3]. R.Balachandran, B.Ramamurthy, X.Zou and N.Vinod Chandran, " CRTDH: An efficient key agreement scheme for secure group communications in wireless ad hoc networks", Proceedings of IEEE international Conference on Communiations(ICC),20(8),pg.1123-1127,2005.

[4] Amir.Y, Kim.Y, Nita-Rotaru.C, Schultz.J, J.Stanton, and Tsudik.G , "Exploring Robustness in Group Key Agreement," Proc. 21st IEEE Int'l Conf. Distributed Computing Systems, pp. 399-408, Apr.2001.

[5] Patrick P.C.Lee, John C.S.Lui and David K.Y. Yau, "Distributed Collaborative Key Agreement and Authentication Protocols for Dynamic Peer Groups," IEEE/ACM Transactions on Networking, Vol. 14, No.2, April.2006.

[6] Steiner.M, Tsudik.G, and Waidner.M, " Key Agreement in Dynamic Peer Groups", IEEE Trans. Parallel and Distributed Systems, vol. 11, no.8, Aug.2000.

[7] Yair Amir, Yongdae Kim, Cristina Nita-Rotaru, John L.Schltz, Jonathan Stanton and Gene Tsudik, " Secure Group Communication Using Robust Contributory Key Agreement", IEEE, vol.15, no.5, May 2004.

[8] William stallings," Cryptography and network security principles and practices",Third Edition, pearson education