Impact Factor (SJIF): 3.632

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

(Special Issue for ITECE 2016)

Analysis of Different Group Key Management Protocols in Mobile Ad-hoc Network

Vishakha Sanghavi[1], Vijay Dubay[2]

[1]*Computer Engineering Department, C.U. Shah University*
[2]*Electronics and Communication Department, School of Diploma Studies, R.K. University*

Abstract — As Mobile Ad-hoc NETwork (MANET) has dynamic infrastructure, poor physical security and shared physical medium, key management is a challenging task. MANET has not fixed structure and so it is of potential security concern because neighbor nodes cannot be trusted. All the group oriented application like video conferencing, software distribution within a group and interactive multi-party business session use the multicast transmission for efficient communication and to save network resources. In many multicast environment, new node can enter and current node can leave at any time and existing members must have to communicate securely using multicast key management for MANET. The much apt solution to provide these services like authentication, confidentiality, integrity and secure multicasting is the establishment of key management protocol.

In this paper, we aim to evaluate an overview of different group key management techniques for MANET like Enhanced Optimized Multicast Cluster Tree (EOMCT) algorithm, Group Member Authentication Protocol (GMAP) and an Efficient Rekeying Function Protocol (ERFP).

Keywords-Mobile Ad-hoc Network; Group Key Management; Group Member Authentication Protocol; Novel Rekeying Function Protocol; Performance efficient EOMCT algorithm

I. INTRODUCTION

Mobile Ad-hoc network (MANET) is a group of independent nodes that communicate with each other, generally with multi-hop scenario. MANET enables easy and immediate communication between two or more nodes without the need of any infrastructure or centralized environment.

MANET is used in wide-scale commercial use, its lack of instinctive security slow down the actual deployment and it becomes the area of interest of many researchers. One more security related concern other than the communication is that the traditional security algorithms like authentication protocols and data encryption techniques cannot be easily adopted in MANET as they have no any certification authority or any fixed key distribution infrastructure is there. In general, a security incident in a MANET does not belong to any individual node, but rather unite to a group like a company or project.[1] The difficult task is how to distribute a key in a group and update a key to ensure secure communication among group members in a MANET such that authenticity, confidentiality and integrity is not violated.

In this paper, we review three different protocols of group key management. EOMCT algorithm provides efficient multicast key distribution with multicast group clustering to solve "1 affects n" problem [2]. Group member authentication protocol [1] is knowledge based and it consists of two phases named group member authentication and secret group key management. A NRFP [5] uses the concept of decentralization group key management and region-based group division. It provides dynamic group membership changes with small computation and storage complexity.

The analysis is organized as follows: Section II to IV briefly review and summarize schemes like Performance efficient EOMCT algorithm, Group member authentication protocol and NRFP applied to group key management in MANET. Section V provides conclusion for this paper.

II. GROUP KEY MANAGEMENT USING PERFORMANCE EFFICIENT EOMCT ALGORITHM

A. Model for EOMCT

This scheme [2] uses a cluster-based hierarchical network topology. Securing multicast key distribution in ad hoc networks is classified into two types: Static clustering in which multicast group is divided into several subgroups. Each subgroup shares a local session key controlled by Local Controller (LC) and Dynamic Clustering which aims to solve "1 affects n" problem.

Optimized Multicast Cluster Tree (OMCT) [6] is dynamic clustering approach to distribute group key on multicast environment. Main aim of this scheme is to find LC for created clusters and it optimizes energy consumption and latency for key delivery. OMCT needs geographical location information of all the group members when key distribution is needed, which does not reflect true connectivity and for cluster information, true connectivity should be considered. To overcome this limit, Optimized Multicast Cluster Tree with Multi Point Relays (OMCT with MPR) [6] is introduced to elect LCs of created clusters using information of Optimized Link State Routing Protocol (OLSR). This scheme assumes

routing control message have been exchanged before key distribution and transmission does not be acknowledged resulting more retransmission, more energy consumption and more delay.

In this technique, OMCT algorithm is enhanced by using Destination Sequenced Distance Vector (DSDV) routing protocol because DSDV maintains routes through periodic and event-triggered exchanges as node joins or leaves avoiding routing loops. Each node has unique sequence number which updates periodically. It reacts quickly on changes in topology and sends acknowledgement to reduce retransmission. By this, average latency and energy consumption is reduced.

B. EOMCT Algorithm

EOMCT which uses DSDV routing protocol to elect LCs of created clusters. The principle of this scheme is to start with group source Group Controller (GC), to collect its 1-hop neighbors using DSDV and to elect LCs which are members having child nodes at next level. In next step, elected LCs cover members having 2-hops neighbors of group source and continue with this steps, until LCs cover all group members. The EOMCT algorithm [2] is described as follows:

Algorithm: EOMCT (clusterhead)

```
Step 1:
ListLCs = ClusterHead
Listnodes = {1,2,3,…, c} // c=number of cluster members
Step 2:
for ( i = 1 to Listnodes) do
  if ( Listnodes ≠ null) then
        if ( i is multicast group) && ( i has group members  childs ) then
                    ListLCs = ListLCs ∪ {i}; // Add i to LCs list
       Listnodes = Listnodes;
       // {group members covered by i};
       // Remove members which are covered by i
       EOMCT (i); // recursively execute algorithm for i
     end if
  end if
end for
Step 3:
if ( Listnodes  ≠ null ) then
  for ( j = 1 to Listnodes ) do
    Compute the reachability factor of  j: number of members    in Listnodes, in 1-hop from node.
  end for
  while ( Listnodes = i ) do
  ListLCs = Listnodes {i} //LC joins new member list
  ListLCs  ≠  Listnodes {i};
  end while
end if
```

EOMCT algorithm is more performance efficient than MOCT because it uses DSDV routing protocol. The example of EOMCT is shown in figure 1. Group source GC 0 collects its 1-hop neighbors by DSDV, and elects node 1 and 7 as LCs, which are group members. Node 1 has child members as 2, 3, 4, 5, 6 and node 7 has child members as 8, 9, 10, 11, 12.

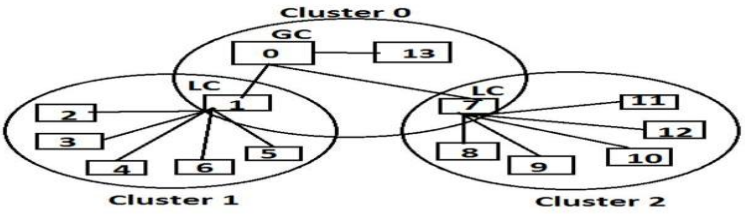

Figure 1. EOMCT[2]

Now if node 13 joins the group, according to step 3 of algorithm, this node is attached to cluster 0 based on connectivity information using DSDV. LC will be selected iteratively until all group members are covered.

C. Analytical model and simulation results for EOMCT

1) Analytical model: Performance of average latency of key distribution and energy consumption is evaluated by analytical model. Average latency can be defined as transfer delay which occurs during transmission of keys from source to destination which is calculated by:

$$AL(N) = t_s + t_k + \sum_{i=1}^{n} ((1 - p_i) N_i).$$

Where, AL = Average Latency, N = No. of packets sent from source to destination, t_s = setup time, t_k = key size, p_i = packet drop, N_i = No. of packets transmitted in the path.

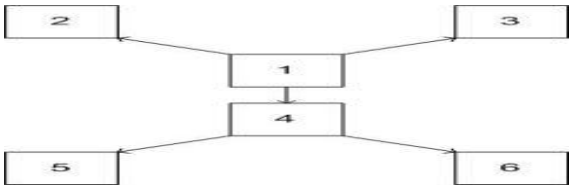

Figure 2. Multicast key distribution

For figure 2, energy consumption is calculated as follows:
E_{MAX} = MAX [$E_{1\text{-}2,}$ $E_{1\text{-}3,}$ $E_{1\text{-}4}$] + [$E_{4\text{-}5}$] + [$E_{4\text{-}6}$].
E_{MIN} = MIN [$E_{1\text{-}2,}$ $E_{1\text{-}3,}$ $E_{1\text{-}4}$] + [$E_{4\text{-}5}$] + [$E_{4\text{-}6}$].
Error = [E_{MAX} + E_{MIN}].
Where E_{MAX} is maximum energy level, E_{MIN} is minimum energy level and Error is total energy level.
2) Simulation: The EOMCT algorithm is simulated under Linux Fedora using Network Simulator (NS)-2 [7] version ns-allinone-2.33. parameters considered in this scheme is network surface (1000m × 1000m, 1500m ×1500m, 2000m × 2000m), density of group members numbers are taken as 7, 13 and 28. Maximum speed of member is set to 10km/h (2.77m/sec), pause time is 20 seconds, simulation duration is 200 seconds, MAC layer is IEEE 802.11, mobility model is random waypoint model with pause time of 20 seconds and with maximum node movement speed of 3m/s, routing protocol is DSDV. Mobility scenarios are generated by *setdest* provided in NS-2. Only unicast distribution of leys exists in simulations.

Table 1. Simulation results Of EOMCT And Omct OMCT With MPR Algorithm

Surface	Nodes	Latency (ms)		Energy (100J)	
		EOMCT	OMCT with MPR	EOMCT	OMCT with MPR
1000	7	0.25	0.5	46	60
×	13	0.5	0.8	55	70
1000	28	0.58	1.2	58	79
1500	7	0.2	0.3	47	63
×	13	0.4	0.64	50	70
1500	28	0.6	0.89	58	86
2000	7	0.12	0.12	50	82
×	13	0.32	0.92	55	86
2000	28	0.51	1.00	60	90

Table 1 shows the simulation results of EOMCT and OMCT with MPR performance for latency for key distribution and energy consumption with surface 1000m× 1000m, 1500m× 1500m, 2000m× 2000m each with nodes density of 7, 13 and 28. From that, it can be observed that EOMCT algorithm performs better for latency for key distribution and energy consumption than OMCT algorithm with MPR.

III. GROUP KEY MANAGEMENT USING GROUP MEMBER AUTHENTICATION PROTOCOL

A. Protocol Design and Communication Model

This technique uses two phases to establish secure group communication in MANET. First phase is knowledge-based group member authentication which is based on Zero Knowledge Proof (ZKP) [3] and second phase is secret group key management which uses Threshold cryptography [4].

First phase is group member authentication which uses secret group key to identify the group that a mobile node belongs to and then recognizes the criteria of group membership status. When a node wants to become a new group

member, it searches for legitimate group members and tries to communicate with them. Based on the node's knowledge, group members compare it with pre-defined "required group membership" and evaluates whether it can join or not. This protocol employs ZKP algorithm, which gives a method to verify a secret key without disclosure of any secure information. In ZKP, a new node behaves as *prover* and legitimate group members behave as *verifiers.*

Second phase is secret group key management which uses threshold cryptography. The secret group key is divided to n shares and later generated by a new group member by the response of t group members. Protocol will be robust because it does not need a key server so it can work with busy MANET. When a node obtains a new secret group key, the protocol recognizes that the new node joins the new group and increases its current group membership and node's knowledge.

In this protocol, three entities are are used: (1) *new member* which is a mobile node and tries to become a group member, obtains shares from t *share holders* and generates a secret group key. (2) *Share holders* which are a set of legitimate users.(3) *Dealer* exists in the group initialization phase. Communication model shows the flow between new member and share holders as given in steps and figure 3[1].

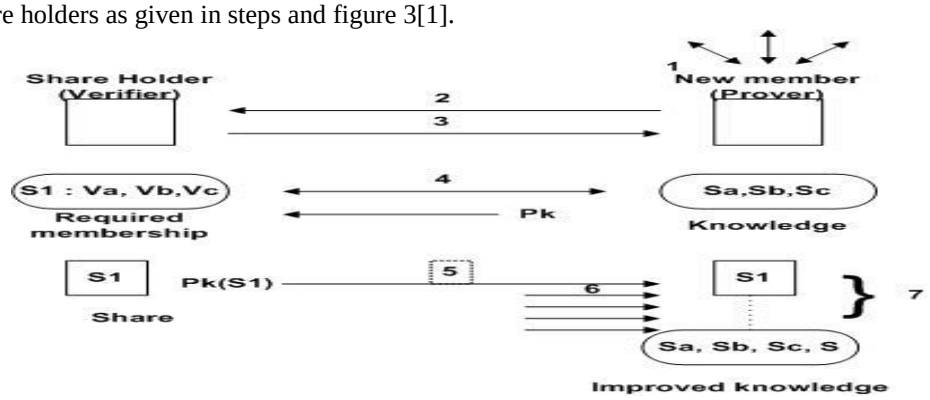

Figure 3 Communication flow between verifier and prover

1. New member discovers IP addresses of share holders for a secret group key S by a share holder discovery procedure. When new member joins a group, it multicasts SHARE_DISCOVERY message to multicast address and when share holder finds the request message, it unicasts SHARE_REPLY message to new member with required group membership and t.
2. New member sends the request to the selected share holder to retrieve share for S.
3. Share holder sends back acknowledge.
4. New member and share holder start challenge and response for a ZKP session to verify whether the new member fulfills the required group membership. Share holder obtains the new member's public key P_k. This large number of challenge and response messages can be performed parallel, making public and private information be a set of quadratic residues modulo N. Vector of knowledge hence aims to speed up protocol by reducing the number of ZKP messages.
5. Share holder gives its share to the new member securely. However, malicious nodes may collect more than t shares and finds the secret group key if share holder sends it over non-secure channel.
6. All above procedures are repeated t times with different share holders.
7. New member finally generates S from t shares by threshold cryptography.

In this protocol, steps 2 to 5 identifies the timeout value s seconds. If share holder leaves from the network within communication then new member cancels the procedure with selected share holder and restarts it with another share holder.

B. Experimental Results of Group Member Authentication Protocol

This protocol is implemented in JAVA with JAVA 2 SDK 1.4.2_05 with 40 classes. Authors needed to connect each machine through 802.11 b BSS mode in a dedicated wireless network.

1) *Setup*: This step consists of (1) group initialization and (2) node initialization steps. First step includes required group membership configuration, a secret group key initialization and (n,t) definition for threshold cryptography. Dealer keeps these, computes polynomial and distributes share and required group membership to n share holders. Second step is required only once for each node. It prepares initial knowledge which is used by a node and is improved a later time.
2) *Evaluation*: in this protocol, two tests are performed. First test is to find the average time of secret group key acquisition from t different share holders in (n,t) threshold cryptography. From this test, it can be shown that Vector of knowledge or ZKP procedure takes less than 10% of total response time but data transmission on wireless link possesses the highest ratio. Second test are performed on data communication cost to complete this protocol. it shows that bit strings for ZKP accounts for approximately 90% of total size and would be propositionally increased to thresholding value. So the trade-off is between protocol robustness and performance.

III. NOVEL REKEYING FUNCTION PROTOCOL FOR MULTICAST KEY DISTRIBUTION

A. Region-Based Group Key Management Model

The proposed protocol [5] partition a group into region-based subgroups and it uses decentralized key management principles [8] to improve scalability of the scheme. Here one assumption is taken that each node is outfitted with Global positioning system so each node knows the location of others while moving across region. For security of group communication and for security of group member communication, secret group key K_G and secret shared region key for region i KRi is used respectively. The average number of nodes in the region are N= λpA where λp is node density and A is operational area of radius r. In random distribution, probability of any node in the group is λ/(λ+μ) and probability of any node that is not in the group is μ/(λ+μ) because node can leave the group at rate μ and can rejoin the group at rate λ. Based on these rates, aggregate join $A_J = \lambda \times N \times \lambda/(\lambda+\mu)$ and aggregate leave $A_L = \mu \times N \times \lambda/(\lambda+\mu)$ can be calculated.

Protocol will work in 3 steps: (1) *Bootstrapping:* in this step, node with smallest id will become a leader of group and implement Group Diffie Hrllman (GDH) and generate a secret leader key K_{RL} to generate a secret group key K_G because it can be calculated like $K_{G=}$ MAC($K_{RL,}$c) where MAC is secure hash function and c is fresh counter. This key is distributed among their group members by leaders of each region. (2) *Key Management*: Key should be managed by doing rekeying if any event occurs for maintaining secrecy. K_{RL} is rekeyed when there is any change in leader. The regional key K_R is rekeyed when regional membership changes. (3) *View Management*: This protocol allows to maintain membership for consistency. Regional View(RV) contains regional member's id and their locations. Leader View(LV) contains leader's id and their locations. Group View(GV) contains group's id and their locations.

B. Novel Rekeying Function Protocol (NRFP)

Three keys and one function is needed for this protocol which are described follow: (1) *MasterKey (MK)*: Each node is stamped with this globally shared key MK which is used by base station (BS) to encrypt the broadcasted message and one local administrative function (LAF). (2) *Local Key(LK):* each node has this key shared with BS for secure communication between them and for rekeying purpose. (3) *Session Key(SK):*It is shared with immediate neighbors of the node and used for secure communication when privacy and source authentication is required.(4) *LAF*: It includes 'master function', 're-keying function' and 'derivation function' for node-to-node secure communication. It is responsible for key generation.

Re-keying function is needed for 2 reasons: (a) It is simple to compute f(k) for k but computationally infeasible to find k for f(k). (b) $k_0, k_1, k_2, \ldots k_n$ is computationally infeasible to compute f(k), as it is computationally infeasible to compute k.

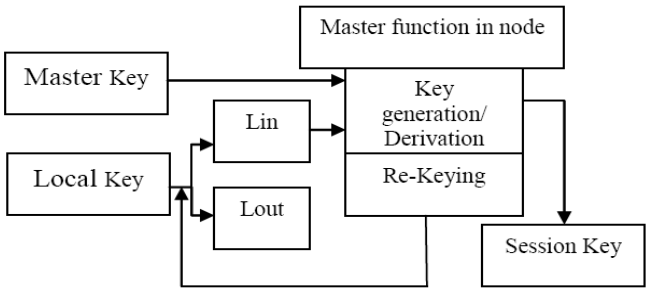

Figure 4. Novel Rekeying Function Protocol [5].

Function and keys stamped through fundamental principles of key management as following:
(1*) Key deployment*: Each node is stamped with unique ID, MK, LK and master function which generates and regenerates the unique sharing key with other nodes deriving from MK and LK and they never exchange. Only SK is exchanged to establish communication.
(2*) Key Establishment*: after deployment, cluster head generates SK and sends the control message to its members to generate SK. When 2 nodes want to communicate with each other, they use same SK to exchange the data securely.
(3) *Node addition*: when a new node enters, it goes for cluster beacon. If it founds the same, key refreshment is done into cluster and generates its own SK. If it dose not found any cluster beacon, it becomes its own cluster and act as a CH and runs LAF to generate keys.
(4*) Node Eviction*: if any node leaves the region, author has proposed 2 cases for this:
Case 1: CH sends a hello message to the node who wants to evict and waits for a certain time. If it does not get reply then it sends a message to all to nodes to inform to delete the nodes.
Case 2: When a CH itself wants to evict then first of all it sends the message about eviction to all other nodes so that other nodes can choose CH based on highest number of a list of neighbors. But if a CH left stealthily, member will not get any beacon for certain time and nodes will elect new CH.

For authentication, this algorithm uses HMAC function. A sender will send the encrypted message e(M) and then resend e(M||K) where k is key. Receiver decrypts M by d(e(M)) to generate M'. (M'||K) are then encrypted and compared with e(M||K). If they match, it is confirmed that data is not corrupted.

NRFP satisfies the following properties: (1) only authorized users can communicate as assigned key material can not be reached to unauthorized users. (2) Session keys are distributed securely because it is based on personal key sharing. When a network is revoked, it cannot recover the session key because it is self generated. Because of broadcast, an outside attacker cannot masquerade as a base station disseminating a session key and start a revocation attack.

C. MDS Code-based Rekeying on a Key Tree

Maximum Distance Separable Codes are a class of error control codes which works on a theory that if GF(q) be a finite field with q elements [9], an (n,k) error control code is then a mapping from $GF(q)^k$ to $GF(q)^n$: E(m) = C_r where m=m1,m2,…mk is original message and c= c1,c2, …cn is code word block. NRFP is focused on the adaption of basic scheme for rekeying on a tree from bottom-up.

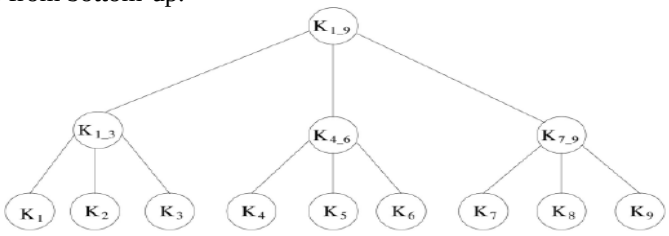

Figure 5. A key tree for a five member group [5].

The GC stores all the pairs of key on key tree. When encryption is required for rekeying, new MDS code is constructed from all key pairs ((j_i s_i ' S) of corresponding immediate child and then multicast by GC. In figure 5, when node 9 leaves, GC first uses group key pairs K7= (j_7, S_7) and K_8 = (j_8, S_8) together with fresh random r to construct a code word of an (L,2) MDS code and then follows the rekeying procedure of basic rekeying scheme. After decoding, node 7 and share new key K_{7-8} and GC constructs code word of an (L,3) MDs code from keys $K_{1-3,}$ $K_{4-6,}$ K_{7-8} and it will generate new group session key $K_{1-8,}$

D. Experimental Results

Results are taken to identify the optimal regional size which will minimize the traffic generated while satisfying security properties in terms of secrecy ,availability and survivability. For that authors have calculated following performance measures:

(a) Group join/leave Cost: It is the cost for handling group join/leave events and cost caused by connection/disconnection events.

$C_{join/leave,i} = |\_A_J \times C_{Join,i} \_| + = |\_A_L \times C_{Leave,i} \_|$

Where A_J and A_L are aggregate join and leave rates of nodes.

A group join requires the C_{intra} which specifies the update of regional view and rekeying of regional key of that region from which join is started and $C_{group,i}$ which specifies update of group view and rekeying of a group key.

$C_{join,i} = |\_ C_{intra} \_| + = |\_ C_{group,i} \_|$

A group leave also includes 2 cases, when a non-leader member leaves and leader leaves the group. So, group leave cost is

$C_{Leave,i} = C_{NonLleader\ Leave,i} + C_{Leader\ Leave,i}$

(b) Group Communication Cost : It includes the group communication between members. It is assumed here that all members are interested in other member's all published data which is at the rate of λ_{pub} . Aggregate rate of published data of each node is:

$A_{pub} = N \times [\ \lambda/(\lambda+\mu)\ ] \times \lambda_{pub}$

Published data can be distributed to all leaders first and then leader will broadcast it to its neighbors in the region.

$C_{GC,I} = A_{pub} \times ((\ N_{region,i} \times M_{pub} \times H_{region}) + (M_{pub} \times H_{region,i}))$

(C) Evaluation of Proposed Cryptographic Scheme:: To evaluate the scheme, multicast key distribution scheme is implementes with 128bits session key among 3-ary balanced key tree and compared with previous cryptographic techniques. Comparison is based on one member departure from 3 members. For this evaluation, MD5 algorithm [10] is used as an ideal hash function which produces 128 bit output for any length of input.

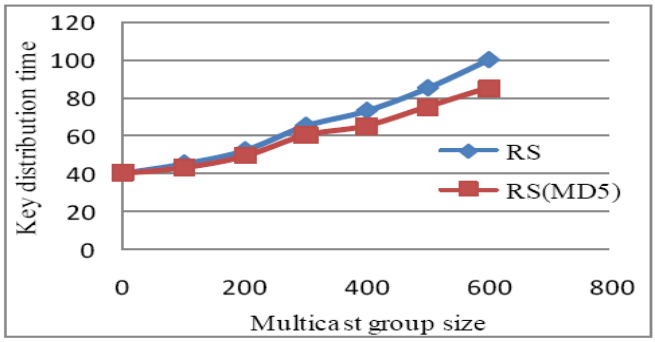

Figure 6. Analysis of NRFP for computation time for key distribution in group [5].

Comparison of computation time of the key dissemination and recovery using different schemes is shown in figure 6. It is observed that one way hash function adds none-trivial computation complexity. Proposed scheme performs better than conventional cryptographic scheme.

Table 2. Computation time comparing To Rc4 approach

Time (us)	RS	RS(MD5)	RC4
GC	19	28	227
Member	2	5	61

Computation time of key distribution is compared with conventional stream cipher RC4 in Table II. And it is observed that computation time is larger when we use RC4 because its key scheduling process has dominant effect.

IV. CONCLUSION

Security of key management for group communication within MANET is an challenging issue. In this paper, we analyzed three key distribution schemes on different paradigm. Analysis of EOMCT algorithm shows that this scheme is performance efficient and suitable for secure multicast key distribution in MANET. Analysis of GMAP protocol uses Zero Knowledge Proof for group member verification and Threshold Cryptography for managing a secret group key in MANET to provide security. Analysis of NRFP protocol shows that it supports source authentication without precluding in-network processing and computation complexity is reduced by employing MD5 rather than using expensive cryptographic function. For maintaining low balanced communication and storage complexity, much lower computation is needed.

REFERENCES

[1] H. Asaeda, M. Rahman, M. H. Manshaei, Y. Fukuzawa, "Implementation of Group Member Authentication Protocol in Mobile Ad-hoc Network", IEEE Wireless Communication and Networking Conference, vol. 4, pp. 2205-2210, 2006.

[2] D. Suganya Devi and G.Padmavathi, "Performance Efficient EOMCT Algorithm for Secure Multicast Key Distribution for Mobile Ad hoc Networks", IEEE International Conference on Advances in Recent Technologies in Communication and Computing, pp. 934-938, Oct. 2009.

[3] U. Feige, A. Fiat and A. Shamir, "Zero Knowledge Proof of Identity", Proc. The 19th ACM Symp. On Theory of Computing, pp. 210-217, May 1987.

[4] Y. Desmedt, "Some Recent Research Aspects of Threshold cryptography", Proc. Information Security, pp. 158-173, Springer-Verlag, 1997.

[5] N.Vimala, B. Jayaram and Dr. R. Balasubramanian, "An Efficient rekeying Function Protocol with Multicast Key Distribution for Group Key Management in MANETs", International Journal of Comuter Applications, vol. 19, issue. 2, pp. 44-51, April 2011.

[6] M.Bouassida, I. Chrisment, and O. Festor, " Efficient Clustering for Multicast Key Distribution in MANETs", LCNS 3642, pp. 138-153, Jan 2008.

[7] K. Fall and K. Vardhan, "The Network Simulator (ns-2)". Available at: http://www.isi.edu/nsnam/ns.

[8] Jin-Hee Cho, ―Design and Analysis of QoS-Aware Key Management and Intrusion Detection Protocols for Secure Mobile Group Communications in Wireless Networks, Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University.

[9] R.J. McEliece and D.V. Sarwate, ―On Sharing Secrets and Reed-Solomon Codes,‖ Comm. ACM, vol. 26, no. 9, pp. 583-584, Sept.1981.

[10] B. Schneier, Applied Cryptography, second ed. John Wiley & Sons, 1996.

[11] V. Sanghavi, S. Sanghvi, N. Tada and V. Dubay, "Analysis of Different Key Distribution Schemes for Mobile Adhoc Network", International Conference on Engineering, NUiCONE 2012, pp. 1-6, Dec. 2012.