Journal of Computer Science 10 (7): 1281-1290, 2014
ISSN: 1549-3636
© 2014 Science Publications
doi:10.3844/jcssp.2014.1281.1290 Published Online 10 (7) 2014 (http://www.thescipub.com/jcs.toc)

AN EFFICIENT AND VARIABLE INTERVAL REKEYING FOR DYNAMIC AD HOC MULTICAST GROUP USING KEY PATH REDUCTION

[1]R. Pugalenthi and [2]T.V. Gopal

[1]Department of Computer Science and Engineering, St. Joseph's College of Engineering, Chennai, India
[2]Department of Computer Science and Engineering, Anna University, Chennai, India

Received 2014-01-08; Revised 2014-02-11; Accepted 2014-02-26

ABSTRACT

Key distribution is one of the major issues in secure ad hoc multicast group communication. There has been an extensive research on rekeying, to reduce cost. In this study, we propose an efficient and scalable batch rekeying for dynamic ad hoc multicast group, with variable interval and key path reduction techniques. The proposed scheme overcomes the major existing issues like inefficiency in using the keys that are generated and distributed, the sync issue in which a user tries to decrypt a data using an irrelevant key, imbalance in network traffic and latency in the key server response to the user request, for leaving and joining at once. Generally, a central key server is used to govern all the above issues. The proposed scheme excludes the usage of a central key server by generating the group key in the individual nodes, which minimizes the communication overhead and the number of keys that each user possesses. The proposed scheme also reduces the depth of the tree very effectively, when a user joins or leaves the group, using key path reduction technique and ensures forward and backward secrecy. The simulation result shows better performance when compared to individual, regular batch and periodic rekeying.

Keywords: Key Distribution, Batch Rekeying, Key Path Reduction, Variable Interval, Ad Hoc Network

1. INTRODUCTION

Multicasting is an efficient method of communication, while delivering services from a sender to a large group of users and it provides proficient and best-effort delivery service (Wong *et al*., 2000; Caronni *et al*., 1998). The service provider must ensure that authorized members alone access the service, by using an access control mechanism, which authenticates the users or ensures that only the authentic users access the service (Wallner *et al*., 1999; Pietro *et al*., 2003). Symmetric key cryptography (Wang and Bhargava, 2005) was the conventional access control mechanism used in group communication with a pre-shared key, otherwise called, group key (Wei *et al*., 2007; Steiner *et al*., 2007; Desmond *et al*., 2006). A group key must be distributed by the key server to all the users of a particular multicast group and when there is any change in the membership, the pre-shared key must be updated and distributed to all the members of the group, which is usually called 'rekeying' (Ballardie, 1996). The two important properties associated with rekeying, are forward secrecy and backward secrecy (Judge and Ammar, 2003; Canetti *et al*., 1999). Any rekeying mechanism must ensure both forward and backward secrecy.

Individual rekeying with Logical Key Hierarchy (LKH) (El-Zoghdy *et al*., 2011) was proposed as a solution for rekeying; it reduces the rekey communication cost considerably (Muthusamy *et al*., 2013). In rooted key tree for a group of N members with the degree of d, where N = 9 and d = 3, the numbers of encryptions by the key server are (Wei *et al*., 2007) Equation (1):

Corresponding Author: R. Pugalenthi, Department of Computer Science and Engineering, St. Joseph's College of Engineering, Chennai, India

$$C_{all} = C_j + C_l = (2 + d) * \log_d N - 1 \quad (1)$$

where, c_j and c_l are the number of rekey messages when a member joins and leaves respectively. However, individual rekeying with LKH has some significant issues, like inefficiency in using the keys that are generated and distributed and the sync issue. The above issues were considered as a waste of server cost. Batch rekeying (Zhang *et al*., 2001; Yang *et al*., 2001) was proposed as a solution to the above issues, in which the server collects the entire join and leaves requests during a particular period of time and processes them. In rooted key tree for a group of N members with the degree of d, where N = 9 and d = 3, the number of rekey messages is (Wong *et al*., 2000) Equation (2):

$$C = N_j * (1 + \log_d N) = N_l * ((d - 1) \log_d N) \quad (2)$$

where, C is the total number of rekey messages and N_j and N_l are the total number of rekey messages when a group of members joins and leaves respectively. The three major issues that arise in batch rekeying are: First, when there is a maximum (N) request to the key server, the network traffic increases in abundance and if the number of requests is 0, the key server lies idle. Second, latency in the key server response to the user request and third, user requests for leaving and joining cannot be responded to at once. To solve all the above issues, we devised an efficient and scalable batch rekeying for dynamic ad hoc multicast group (Kaya *et al*., 2003; Zhu *et al*., 2003) with variable interval (Prathap and Vasudevan, 2008) and key path reduction techniques. Our experimental results show that the proposed scheme is very efficient when compared to the existing individual, regular batch and periodic rekeying approaches. The rest of the paper is organized as follows. Section 2, presents the related works. Section 3, are presents the proposed protocol description. In sections 4 and 5, we present the proposed join and leave protocol respectively. Section 6, analyzes the results of the proposed protocol and section 7 concludes the paper.

2. RELATED WORKS

In this section, different approaches used for rekeying are discussed; they are: Batch rekeying with variable intervals and rekeying using key path compression.

2.1. Batch Rekeying with Variable Intervals

In this approach (Jin *et al*., 2002; Li *et al*., 2001), an analysis was done using the user's request and response delay in batch rekeying. Two major factors were considered; the average delay of the user's request-response and the number of users' requests in the batch interval. Both the factors have a substantial impact on the rekeying cost. A long average delay of the user request-response will reduce the system's safety and will also reduce the interest of users in multicast services and increase the cost of rekeying. The number of user requests which is influenced by the batch interval, also has a great effect on the rekey traffic. The result of the analysis (Jin *et al*., 2002) shows that there is a delay in the response to the user's request, which is relevant to the batch interval and the parameter of the distribution. The batch rekeying with variable interval is more suitable for the network, than the fixed interval and any network that wants steady rekey traffic must satisfy the following inequality Equation (3):

$$\Delta t \geq -\frac{1}{\lambda} \ln\left(1 - \frac{d}{N} \exp(\lambda t_o)\right) \quad (3)$$

Where:

d = The number of user requests in the Interval $t(t_0, t_0 + \Delta t)$

N = The total number of user requests in the Interval $(t_0, t_0 + \Delta t)$

t = The time duration which ranges from 0 to t

2.2. Rekeying using Key Path Compression

In this approach (Raju *et al*., 2010), the members are organized into a logical key hierarchy and use both symmetric and asymmetric encryption for key distribution (Raghini *et al*., 2013). This approach dynamically selects a member as a sponsor, to reduce the chances of the generation of a weak group key, uses neither any central server to update the group key nor secure channels for transmitting the keys.

When a member wants to join a group, it broadcasts a join request along with its public key. The join request reaches the join point and the sponsor node of the tree. All the nodes except those who have a root as their parent, along the co-path of the join point are detached from the key path of the join point and attaches themselves to the root, thus the path is compressed. The join point parent node also attaches itself to the root. Two new nodes will be created by the existing group members and added as the children to the join point. The sponsor associated with the join point will be given to the right child node of the join point and associate the new member

and its public key with the left child node of the join point. The sponsor and the root exchange the new group key. The sponsor encrypts the new group key with the old group key and broadcasts and also sends the new group key and the updated public keys of the key path nodes to the new member encrypted with the public key of the new member. All the group members update their public and private keys and also the public keys of their ancestor nodes.

When a member wants to leave, it broadcasts the leave request. The Sponsor of the sub-tree rooted at the leaving member parent will act as sponsor. The nodes and members, who are in the co-path of the leaving member, send their private keys to the root encrypted with the root's public key and detach themselves from the key path of the leaving member and attach themselves to the root directly. The intermediate nodes along the co-path of the insertion point, who have the root as their parent, remain the same. Both the root and the sponsor get the group key, using the Diffie-Hellman algorithm. The root will encrypt the new group key using the private keys and rearrange the tree.

3. PROTOCOL DESCRIPTION

This part describes the proposed scheme, which is an efficient and scalable batch rekeying for the dynamic ad hoc multicast group with a variable interval and key path reduction. The proposed scheme is divided into two major parts, namely, batch rekeying with variable interval and rekeying using key path reduction.

3.1. Batch Rekeying with Variable Interval

The idea of the batch processing of individual rekeying comes from the fact that the membership changes in a multicast group may happen at almost the same time. The batch processing suggested in this study uses two important mechanisms, namely, Interval Manager (IM) and Request Controller (RC). The general operations of the IM and RC are; whenever a user needs to join or leave an ad hoc multicast group, it broadcast a join or leave a request. The RC collects all the join or leave requests over a period of time. The time duration for collecting the user request, either a join request or a leave request is called an interval. In conventional batch rekeying (Wong *et al*., 2000), a fixed interval would be maintained to collect the user's join and leave requests, which leads to an increase in the network traffic when there are more requests, or the key server would be idle when there are no user requests during a particular time interval. In the proposed scheme, the Interval Manager (IM), dynamically fixes the time duration of the interval based on real-time factors, like the number of requests over a particular time, the average delay of the users' request-response, the traffic network overhead and latency. The number of requests over a particular period of time can be initialized by the administrator in the IM, based on the nature of the multicast application in which the proposed scheme is implemented. Other factors like the average delay of the users' request-response, t he network traffic and latency will be considered by the IM, to compute the time interval for every batch of user requests. As controlled by the IM, the RC holds the entire join or leaves requests for a particular duration $(t_0,t_0+\Delta t)$, where it ranges from 0 to t, computed based on the real-time factors. Let r be the number of requests from users in an interval of $(t_0,t_0+\Delta t)$, with r being variable, the variable batch interval satisfies the following inequality Equation (4):

$$\Delta t \geq -\left(\frac{1}{\lambda}\right)\ln\left(\left(1-r^{(t_0,t_0+\Delta t)}\right)-\left(\frac{d}{N}\right)\exp(\lambda t_0)\right) \qquad (4)$$

Where:
d = The number of requests in $(t_0,t_0+\Delta t)$
N = The total number of user request in $(t_0,t_0+\Delta t)$.

3.2. Rekeying Using Key Path Reduction

The second part of the proposed scheme is rekeying, using the path reduction technique. After collecting all the user requests, the RC forwards them to the Join Point Selector (JPS) in the case of the join request, or to the Leave Point Selector (LPS) in case of leave requests. The general operation of the proposed scheme when a user joins and leaves a dynamic ad hoc multicast group is shown in **Fig. 1 and 2** respectively. The functionality of the JPS and LPS is described as Join protocol and Leave protocol in sections 5 and 6 respectively. The JPS identifies the nodes to which the new users are to be attached. The LPS identifies the nodes from which the users are to be detached. Both the LPS and JPS, rearrange the tree hierarchy after generating the new group key by the individual nodes, using the random number and the member id assigned by the RC.

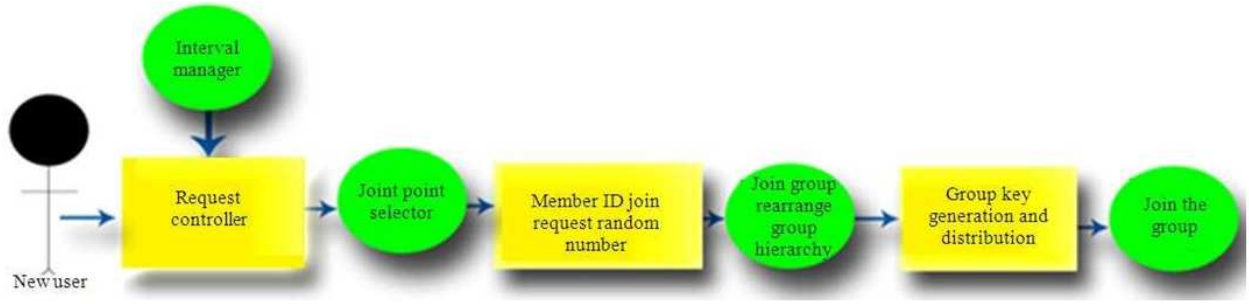

Fig. 1. User join request

Fig. 2. User leave request

4. JOIN PROTOCOL

When a member wants to join a group (M8), it broadcasts a join request to the RC. The RC assigns a unique member id and a random number to all the members who have submitted join requests. The JPS selects a node as the join point, where exactly the new member will be attached in the tree. Usually a join point (2,3) is the right most shallowest leaf node of the tree; in case the right most shallowest leaf node of the tree is not available, two other different criteria will be used to select a Join Point in the tree. If the Join Point is a non-leaf node with one member as a child, then the member will be selected as a Join Point or the sub tree rooted at the sibling node of the Join Point will be selected. Once the join point is selected, other two nodes in the tree are selected as Sponsor-1 (S1) node and Sponsor-2 (S2) node. S1 (2,2) is a node along the co-path of the join point without having the root as a parent and S2 (2,1) is node which is placed exactly as a mirror image of S1 (2,2) in a balanced tree. Both S1 (2,2) and S2 (2,1) are detached from the key path and attach themselves to the root. Four new nodes will be created by the existing group members and will be added as the children of the S1 (2,2) and S2 (2,1) nodes. Thus the key path is reduced on both sides of the tree. **Figure 3** shows tree before join and **Fig. 4** shows tree after join. Once the path reduction takes place and the new member joined the tree, the S1 and the root exchange the new group key. S1 encrypts the new group key with the old group key and broadcasts it. The S1 also sends the new group key and the member id of the key path nodes after the key path reduction, to the new member encrypted with the member id of the new member. All the group members update their member id and also the member id of their ancestor nodes, using the equations, given below:

UMI = HNGK | MI
M7 → JP: EOGK, NGK
M7 → M8: E(MI of M8, NGK, 0,0, 2,3

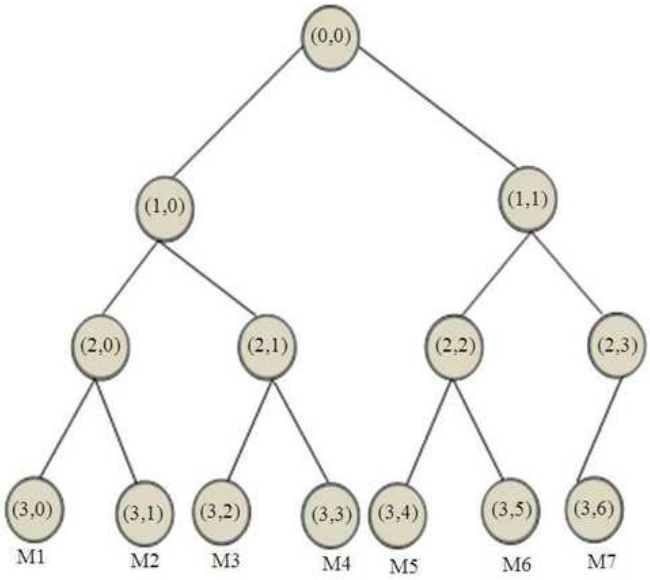

Fig. 3. Tree before join

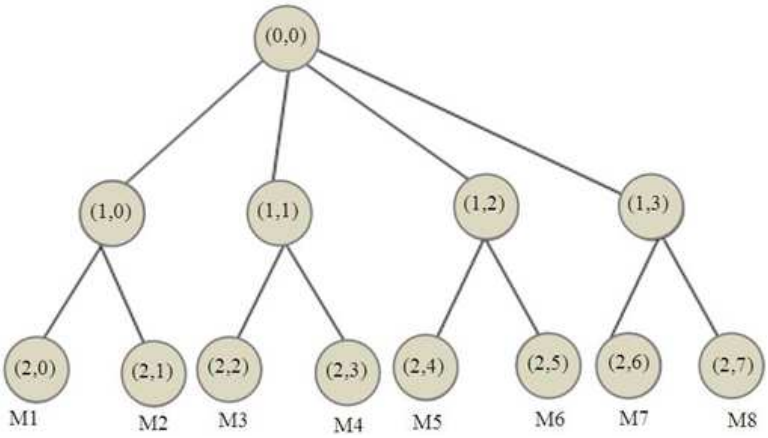

Fig. 4. Tree after join

Where:

UMI = Updated Member Id
H = One-way hash function
MI = Member Id
NGK = New Group Key
JP = Join Point
E = Encryption function
OGK = Old Group Key

5. LEAVE PROTOCOL

When a member wants to leave a group (M8), it broadcasts a leave request to the RC. The member id of all the users who submitted a leave request will be collected and forwarded by the RC to the LPS to select the leave point from where the member would be detached from the tree. Usually, a leave point (e.g., (2,3)) is the right most shallowest leaf node of the tree. Once the leave point is selected, two other nodes in the tree will be selected as the Sponsor-1 (S1) and Sponsor-2 (S2) nodes. The criteria for selecting the S1 (3,6), node are if the member who is leaving does not have any sibling then the S1 of the sub tree rooted at the leaving member parent will act as S1; if the leaving member has a sibling, then it will act as S1. The criteria for selecting the S2 node is, the node along the co-path

of the Leave Point without having the root as a parent and S2 (3,1) is a node which is placed exactly as a mirror image of S1 (3,6) in a balanced tree. Once S1 and S2 are identified, on the right half, the nodes and members excluding S1 who are in the co-path of the leaving member will send their member-id to the root encrypted with the old group key and detach themselves from the root directly. The intermediate nodes along the co-path of the insertion point that has the root as their parent, will remain the same. On the right half of the tree, the nodes and members excluding S2 who are in the co-path will send their member-id to the root encrypted with old group key and detach themselves from the key path and attach themselves to the root. Two new nodes will be created by the existing group members and will be added as the children. Thus, the key path is reduced on the left side of the tree. **Fig. 5 and 6** show the tree before and after leave respectively. The root encrypts the new group key, using the private key and broadcasts it:

(1,0) → (0,0) :E(MI(0,0), UMI(1,0)
(2,2) → (0,0):E(MI(0,0),UMI(2,2)
ROOT(0,0) →GK:E(UMI(1,0),NGK)
ROOT(0,0) →GK:E(UMI(2,2),NGK)

Where:
UMI = Updated Member Id
MI = Member Id
NGK = New Group Key
E = Encryption function
GK = Group Key

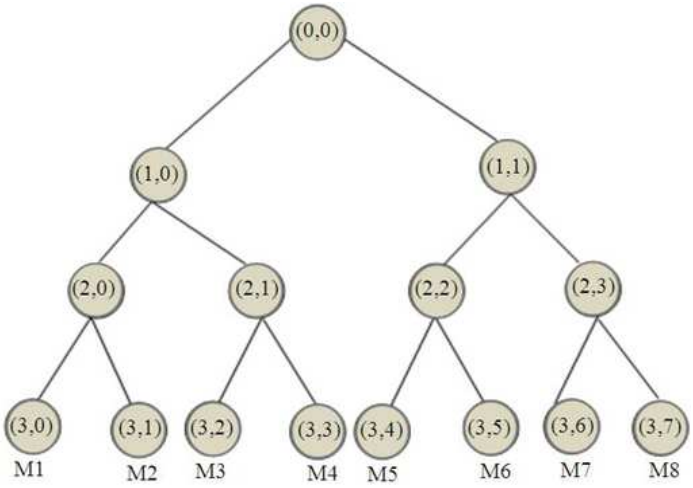

Fig. 5. Tree before leave

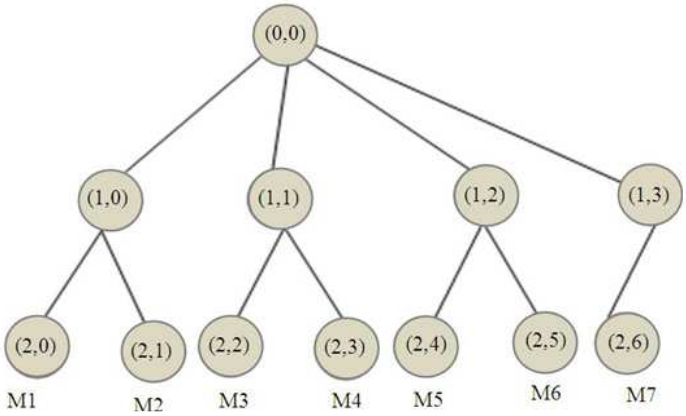

Fig. 6. Tree after leave

6. RESULT AND DISCUSSION

In this section, we examine the performance of our proposed scheme, which was evaluated in the simulation environment. The two major aspects evaluated are, the number of messages generated and the dynamic change in the membership over a period of time.

6.1. Simulation Environment Evaluation

The performance of the proposed scheme was tested using the simulation tool OMNeT++version 4.3, a popular simulation platform among the scientific community. The simulation environment and the underlying topology are shown in **Fig. 7**. In the simulation environment, the performance of the proposed scheme is compared with those of three other well-known schemes, like individual rekeying, regular batch rekeying and periodic rekeying with various parameters. **Table 1** shows the various parameters and their values used in a simulation environment.

6.2. Total Number of Messages Generated

The parameter values given in **Table 1** were used to compare the performance of the proposed scheme with those of individual keying, regular batch rekeying and periodic rekeying.

Figure 8 shows the number of messages, whose values are to be gathered during a time period of 5 min. As the graph shows, the number of messages generated by the individual, regular batch and periodic rekeying are much greater than that of the proposed scheme. The proposed scheme consumes less number of messages.

Figure 9 shows the total number of messages gathered by all the four rekeying schemes with a total of hundred users within the time period of five minutes. The individual rekeying method gathers a huge number of messages among all and generates more number of messages, whereas the proposed scheme generates a much less number of messages. It shows that the proposed scheme is more efficient than the other schemes.

6.3. Dynamic Change in Membership

Evaluating the performance of a protocol during the change in the membership is a major and essential criterion for any rekeying scheme. This section describes the evaluation results according to the dynamic membership states. To evaluate the performance of the proposed scheme during the membership change, the parameter values given in **Table 1** were used in the simulation environment. To adjust the frequency of the users' join and leave, the value of 'duration of leave' has been changed from 5 to 60 min. If the values are less, the group membership is changed more frequently and if the value is more, the group membership is not changed frequently. The comparison is shown in **Fig. 10**. As shown in the graph, the proposed scheme is much more efficient than the other schemes, as the environment has more dynamic changes in the membership. This section describes the evaluation results according to the period. In the simulation environment, the parameter values as shown in **Table 2** were used to evaluate the effect of the period. The value of the period has changed from 2 seconds to 20 seconds, as shown in **Table 3**. The results are shows in **Fig. 11**. As shown in the graph, the gap of the performance is not much, but the proposed scheme is more efficient than the other rekeying schemes.

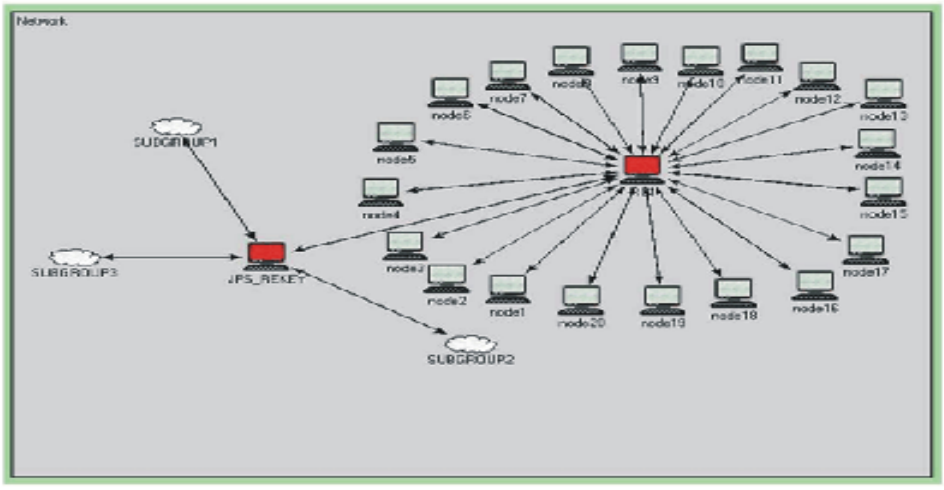

Fig. 7. Simulation environment with OMNeT++

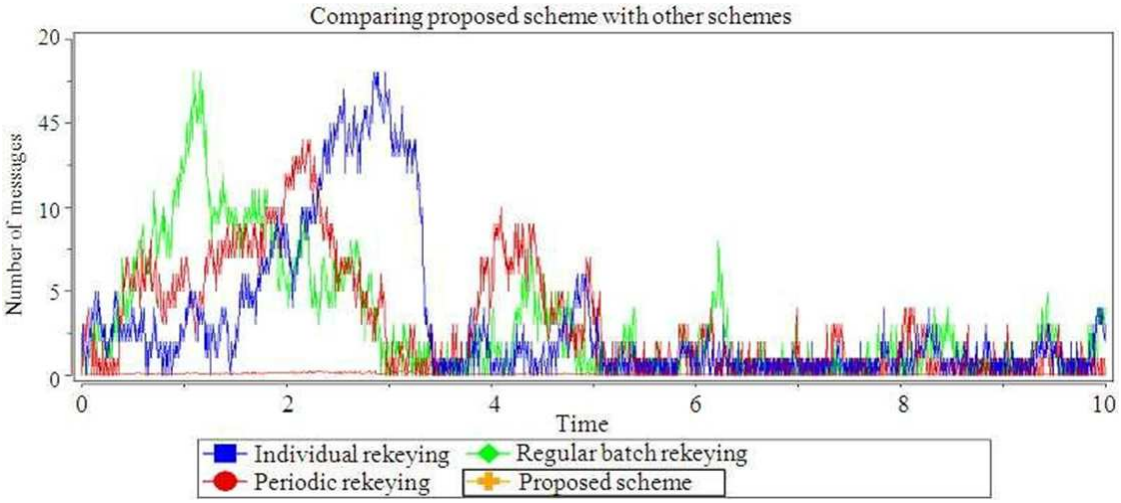

Fig. 8. Comparing proposed scheme with other schemes

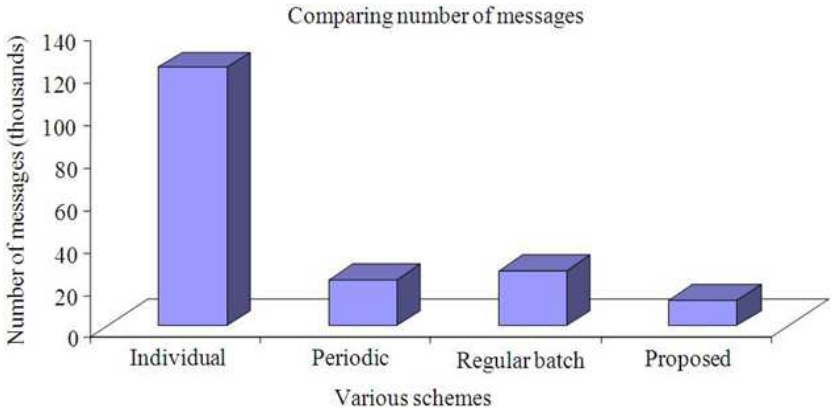

Fig. 9. Comparing the total number of messages in various schemes

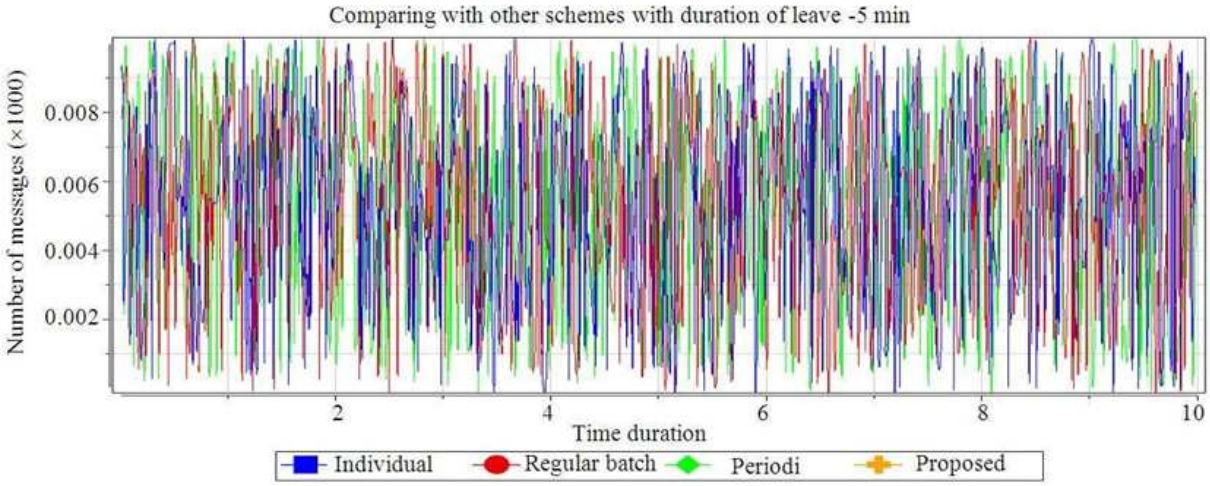

Fig. 10. Comparing with other schemes with 5 min as the duration of leave

Table 1. Simulation environment parameters

Parameter	Values
Evaluation duration	10 H
Number of users	100
Period	20 sec
Duration of Join	5 min
Duration of Leave	5 min
Link Delay	15 ms

Table 2. Simulation environment parameters

Parameter	Values
Evaluation duration	10 H
Number of users	100
Period	20 sec
Duration of join	5 min
Duration of leave	5 to 60 min
Link delay	15 ms

Table 3. Simulation environment parameters

Parameter	Values
Evaluation duration	10 H
Number of users	100
Period	2-20 sec
Duration of join	5 min
Duration of leave	20 min
Link delay	15 ms

7. CONCLUSION

In this study, a new rekeying approach to dynamic ad hoc multicast groups with variable interval and key path reduction technique is proposed. Through the experimental results, it has been proved that the proposed scheme addressed all the major issues that exist in the individual, regular batch and periodic rekeying approaches. The proposed scheme is more suitable for the network, which has steady rekey traffic. The computation and communication overhead can be reduced when highly frequent membership events occur. Our scheme neither requires any central key server to update the group key, nor secure channels for transmitting the keys. Our scheme also addresses various other issues like sync problems, network traffic imbalance, inefficient key usage and user requests for leaving and joining at once. The number of keys and the cost of a join and leave and the experiment are estimated. The result shows better performance when compared to the existing individual, regular batch and periodic rekeying schemes. In future, new modules may be included, which increases the compatibility of the proposed scheme that makes an application much better.

8. REFERENCES

Ballardie, A., 1996. Scalable multicast key distribution. RFC.

Canetti, R., J. Garay, G. Itkis, D. Micciancio and M. Naor *et al*., 1999. Multicast security: A taxonomy and some efficient constructions. Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies, Mar. 21-25, IEEE Xplore Press, New York, NY, pp: 708-716. DOI: 10.1109/INFCOM.1999.751457

Caronni, G., M. Waldvogel, D. Sun and B. Plattner, 1998. Efficient security for large and dynamic multicast groups. Proceedings of 7th IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Jun. 17-19, IEEE Xplore Press, Stanford, CA., pp: 376-383. DOI: 10.1109/ENABL.1998.725721

Desmond, W.H.N., H. Cruickshank and Z. Sun, 2006. Scalable balanced batch rekeying for secure group communication. Comput. Security, 25: 265-273. DOI: 10.1016/j.cose.2006.02.006

El-Zoghdy, S.F., A. Saroit and M. Matar, 2011. A scalable and distributed security protocol for multicast communications. Int. J. Netw. Security, 12: 61-74.

Jin, Q. G. Jianhua, J. Ming and Z. zhangm, 2002. On batch rekeying based on membership dynamics model of multicast. Proceedings of the IEEE Region 10th Conference on Computers, Communications, Control and Power Engineering, Oct. 28-31, IEEE Xplore Press, pp: 145-147. DOI: 10.1109/TENCON.2002.1181236

Judge, P. and M. Ammar, 2003. Security issues and solutions in multicast content distribution: A survey. IEEE Netw., 17: 30-36. DOI: 10.1109/MNET.2003.1174175

Kaya, T., G. Lin, G. Noubir and A. Yilmaz, 2003. Secure multicast groups on ad hoc networks. Proceedings of ACM Workshop on Security of Ad Hoc and Sensor Netw., pp: 94-102. DOI: 10.1145/986858.986872

Li, X., Y.R. Yang, M.G. Gouda and S.S. Lam, 2001. Batch rekeying for secure group communications. In Proceedings of the 10th International Conference on World Wide Web, May 01-05, ACM Press, Hong Kong, pp: 525-534. DOI: 10.1145/371920.372153

Muthusamy, K.S., P. Thiyagarajan and L. Selvaraj, 2013. An enhanced and cost effective group key management scheme for multicast network. J. Comput. Sci., 9: 477-487. DOI: 10.3844/jcssp.2013.477.487

Pietro, R.D., A. Durante and L.V. Mancini, 2003. A reliable key authentication scheme for secure multicast communications. Proceedings of the 22nd IEEE Symposium on Reliable and Distributed Systems, Oct. 6-18, IEEE Xpolre Press, pp: 231-240. DOI: 10.1109/RELDIS.2003.1238073

Prathap, J.P.M. and Vasudevan, 2008. Revised variable length interval batch rekeying with balanced key tree management for secure multicast communications. IJCSNS Int. J. Comput. Sci. Netw. Security, 8: 232-241.

Raghini, M, N. Uma Maheswari and R.Venkatesh, 2013. Over view on key distribution primitives in wireless sensor network. Int. J. Comput Sci., 9: 543-550.

Raju, D.V.N., V.V. Kumar and K.V.S.V.N Raju, 2010. Scalable rekeying for dynamic groups using key path compression. Proceeding of 2nd International Conference on Communication Systems and Networks, Jan. 5-9, IEEE Xplore Press, Bangalore, pp:472-473. DOI: 10.1109/COMSNETS.2010.5431963

Steiner, M., G. Tsudik and M. Waidner, 2000. Key agreement in dynamic peer groups. IEEE Trans. Parallel Distributed Syst., 11: 769-980. DOI: 10.1109/71.877936

Wallner, D.M., E.J. Harder and R.C. Agee, 1999. Key management for multicast: Issues and architectures. National Security Agency.

Wang, W. and B. Bhargava, 2005. Key distribution and update for secure inter-group multicast communication. Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, Nov. 07-10, ACM Press, Alexandria, VA, USA, pp: 43-52. DOI: 10.1145/1102219.1102227

Wei, G.J., H.D. Liang, W. YuMing and Y. ZhongKai, 2007. An efficient rekeying approach for secure multicast communication. Proceedings of the International Conference on Wireless Communications, Networkin and Mobile Computing, Sept. 21-25, IEEE Xplore Press, Shanghai, pp: 1949-1953. DOI: 10.1109/WICOM.2007.488

Wong, C., M. Gouda and S. Lam, 2000. Secure group communications using key graphs. IEEE/ACM Trans Network., 8: 16-30. DOI: 10.1109/90.836475

Yang, Y.R., X.S. Li, X.B. Zhang and S.S. Lam, 2001. Reliable group rekeying: A performance analysis. Proceedings of the Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, Aug. 27-31, ACM Press, San Diego, CA, USA, pp: 27-38. DOI: 10.1145/383059.383062

Zhang, X.B., S.S. Lam, D.Y. Lee and Y.R. Yang, 2001. Protocol design for scalable and reliable group rekeying. IEEE/ACM Trans. Network., 11: 908-922. DOI: 10.1109/TNET.2003.820256

Zhu, S., S. Setia, S. Xu and S. Jajodia, 2004. GKMPAN: An efficient group rekeying scheme for secure multicast in ad-hoc networks. Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, Aug. 22-26, IEEE Xplore Press, DOI: 10.1109/MOBIQ.2004.1331709