Efficient Key Management Scheme for Secure Multicast in MANET

J. Lakshmanaperumal[+], K.Thanushkodi[++], N.M.Saravana kumar[+++], K.Saravanan[++++], D.Vigneshwaran[++++], T.Purusothaman[++++]

+ - Faculty of Electrical and Electronics Engineering, Government College of Engineering, Salem.
++ - Faculty of computer science and Engineering, Akshaya College of Engineering, Coimbatore.
+++ - Faculty of Computer science and Engineering, Bannari amman Institute of Technology, sathy.
++++ - Faculty of Computer science and Engineering, Government College of Technology, Coimbatore.

SUMMARY:

Secure Multicasting (SM) is a popular communication approach in which secure transmission of information takes place from one source to many receivers. The nature of military applications necessitates the use of security features such as confidentiality, source/group authentication etc. Further the mobile ad hoc network (MANET) used in the military applications require these protocols to be implemented in an energy efficient way. In this paper, an efficient way of multicasting a secure data to a group using a hybrid key management scheme is discussed and from the results it is observed that the storage complexity, communication complexity and computation complexity are very much comparable with the existing method.

Keywords:

Boolean Minimization method, Huffman coding, Hybrid key management scheme, Communication Complexity, Computation Complexity

1. INTRODUCTION

Ensuring secure multicast communication involves distributing crypto graphic keys to the members so that only the members of the group can participate in group communications. To establish a safe and secure group communication, a suitable key distribution scheme should be employed. Thus the key management protocol plays a vital role in providing security for multicasting.

The most important aspect of SM is group dynamism i.e., the members of the multicast group can join and leave the group at any time without intercepting the current group. In group dynamics it is mandatory to change the keys for multicast group members for the following two reasons:

(i) A departed or evicted member should not be allowed to receive any further message intended for the group. This is referred as ensuring forward secrecy and

(ii) The previous transactions should not be disclosed to a newly joined member. This is referred as ensuring backward secrecy.

In general, the process of maintaining forward and backward secrecy requires key changing, called group rekeying. Besides changing keys to ensure forward and backward secrecy, it is also required to change the keys periodically to make it difficult for the hackers to trace the keys.

This can be efficiently achieved by combining the key graph method of Boolean minimization technique and Modified Huffman technique and also the efficiency may further be improved by introducing semantic concept in the group controller node.

2. RELATED WORKS

2.1 Boolean Minimization Technique:

In Boolean minimization technique [1,2,7] each user is assigned to a unique key called UserID(UID). The length of the UID is based on the number of users in the group and is calculated as below.

Length of UID = $\lceil \log_2 N \rceil$

where N is the number of users in the group.

The UID can be represented as $X_{n-1}X_{n-2}\ldots\ldots X_0$ where X_i can take values either 0 or 1. The members receive the following two different keys in order to participate in the group.

- Group key – Used to decrypt or encrypt data intended for the group members.
- Auxiliary keys – A set of keys to update the group key in a secure manner.

GK

Manuscript received November 5, 2010
Manuscript revised November 20, 2010

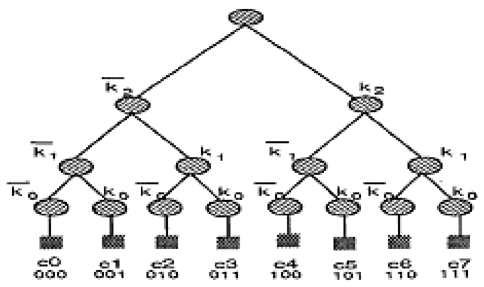


Fig 1. Key tree

The implementation of key management scheme employs a key structure. The sample key tree structure constructed by the group controller with eight users is shown in Fig.1. The auxiliary keys used to manage the session are indicated in the tree structure as k_0, k_0', k_1, k_1', …….k_{n-1},k_{n-1}'. Here the group controller manages all the auxiliary keys { k_0, k_0', k_1, k_1', …….k_{n-1},k_{n-1}'} along with the group key, GK. The leaf nodes indicate as a square box of the tree represent the users in the group. Since there are eight members in the tree, each member is identified by 3 bit UID. The nodes between the root node and the leaf nodes represent the auxiliary keys in the system. Group key GK, which is at the root, is shared by all users. For example, member c7 with UID 111 possesses the auxiliary keys k_2, k_1,k_0 and group key GK.

Members join and leave operations:

2.1.1 Individual Member Removal:

Whenever a member of a multicast group is to be expelled, new group key needs to be disseminated to every member except the one who departed to make sure that the expelled member can no longer send and receive data addressed to the group.

In order to update the new group key GK, the controller has to compute the group key GK_{new} and this is encrypted with the complementary of the auxiliary keys of the departed member.

For example assume the user 5 with UID 101 in Figure1 is leaving from the group. The user posses the auxiliary keys k_0, k_1', k_2 and the group key GK. In order to maintain the forward secrecy, the session key has to be changed and should be encrypted in a such way that the user who left the group should not be able to decrypt it, to accomplish this, a new session key is generated and is sent as 3 different messages encrypted by the three different auxiliary keys that are complementary to the evicted user and its details are given below:

- Auxiliary keys possessed by the evicted user are k_0, k_1', k_2 .
- Complementary to that keys are k_0',k_1,k_2'.
- Three new re-key messages are { E_{k0}'(GK(new)), E_{k1}(GK(new)), E_{k2}'(GK(new)).}

In case the departing user also receives all the messages, it is not possible for him to decrypt the new group key, since every message is encrypted with a new auxiliary key that the departing member does not posses. This scheme also guarantees that every other member of the group can decrypt at least one of the re-key messages. This is due to the fact that the UID of every other member differs from the UID of departing member in at least one bit position, and therefore their key sets are different and can be used for decrypting the new Group key.

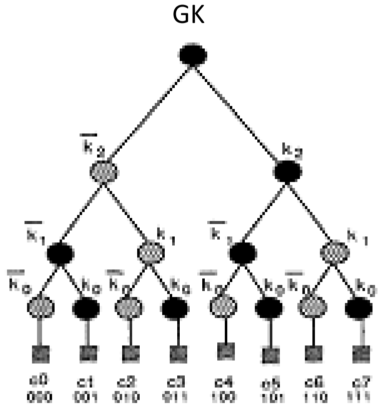


Fig 2. Individual removal

● Signifies the keys that are not used for Re-keying process after departure of c5

▨ Usable keys for other users after the departure of c5

The figure 2 illustrates the re-keying method discussed above. In this figure the keys possessed by the solid nodes are nothing but auxiliary keys and all of them are also with the departed user. Hence, the re-key messages should not be encrypted with these auxiliary keys. For this reason, the complementary of the auxiliary keys are used for encrypting the group key. The hatched nodes symbolizes the complementary set that is the keys not obsessed by c5. The salient point to be noted is that the path from c5 to root has only solid nodes. Every other branch has at least one hatched node on its way to the root.

Now it is ensured that the new group key is encrypted with the complementary set of the leaving user, all members except the member who left the group will be able to decrypt at least one message and hence the new group key is communicated to the remaining members in a secure manner.

2.1.2 Multiple Member Removal:

In practice there are number of situations in which many users may leave at a time , Under such situations there must be a way to provide secure multicasting for only the remaining valid users. The tackling of such a situation is dealt in this section. The key update procedure can be applied k times consecutively to remove k member from the group. However, a more efficient way is to aggregate the removal of several members from the group. This will be useful where several members depart either simultaneously or within very small time interval. The problem of cumulative group removal; becomes grouping the remaining members based on the UID bits in such a way that the members intact are grouped together separated from those who were removed. This is done by grouping the members based on the way they share common bits among one another and in the way they differ in bits with those members who were removed. The multiple removal of users can be dealt with Boolean minimization technique and the same is explained below.

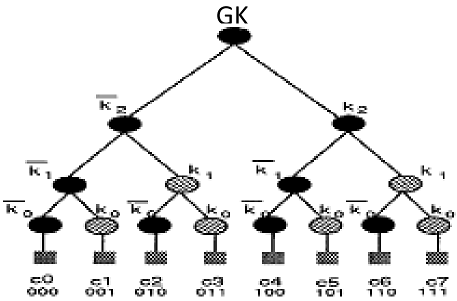
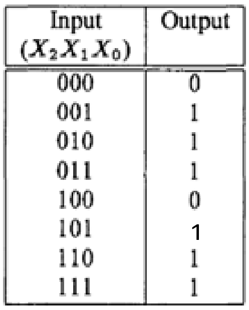


Fig 3. Multiple member removal

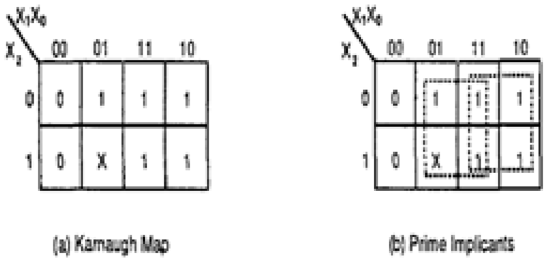
Let us consider the same example as illustrated in figure 3 where two members c0 and c4 are leaving the group. The membership function for the available members is 1 and for the evicted member is 0. The member ship is as shown below

Table 1 Boolean Membership Function



Input $(X_2X_1X_0)$	Output
000	0
001	1
010	1
011	1
100	0
101	1
110	1
111	1

The membership function corresponding to all other members except c0 and c4 is 1 and is illustrated in Table 1. Using this member function, karnaugh map is constructed as shown in figure 4. Each field of the karnaugh map corresponds to a specific minterm and is marked as 0, 1 0r X(for dummy nodes). The next step of the minimization procedure is to identify the largest possible rectangle that contains 1. These rectangles are called prime implicants of the function and by choosing the minimum number of the prime implicants the minimum SOPE of the function is obtained. For this example, the minimization function is (k0+k1) and the new group key is multicasted with the minimization function. It is evident that the left users c0 and c4 does not possess either k0 or k1 but all the other users have either k0 or k1 and hence they can decrypt the new group key. The rekeying message now required is only 2 unlike 6 if the leaves are considered separately.



X_2 \ X_1X_0	00	01	11	10
0	0	1	1	1
1	0	X	1	1

(a) Karnaugh Map

X_2 \ X_1X_0	00	01	11	10
0	0	1	1	1
1	0	X	1	1

(b) Prime Implicants

Fig 4. Karnaugh map minimization of membership function

2.1.3 Join:

Whenever a new member joins the group the centralized server gives the UID to the new member and calculates the new group key. It is first sent to the new member by unicast. It is then encrypted by the old group key and sent to all the remaining members by one multicast. This can be further enhanced by considering the following three scenarios.

a) Number of leave request equal to join request
b) Number of leave request is less than join request
c) Number of leave request is greater than join request

2.2 Modified Huffman's Techniques:

In this Modified Huffman's Technique the UID is given based on the probability of leave of the user. More number of bits is allocated to a member who is having less probability of leave and less number of bits is allocated to a member having high probability of leave.

The group members are arranged by the probability of leaving as given below

User 0 ------0--------k0'
User 1-------01-------k1'k0
User 2-------011-----k2'k1k0
User 3 ------111------k2k1k0.

2.2.1 Members Leave and Join operation:

Members are expected to leave according to their probability of leave. In the above example, the User0 leaves the group first, because its probability of leave is higher than the others. Whenever a member leaves/joins the group the new group key is generated and encrypted by ki+1, ki+1' where 'i' is the most significant bit of the user. and it is multicast to the group. Every other member except the leaving member will be having either k_{i+1} or k_{i+1}'and hence every other member except the leaving member can decrypt the message.

3. MOTIVATION AND CONTRIBUTION

The individual techniques described above may not be suitable for implementing multicast in MANETs. Boolean technique gives better security but the complexity is too high. In Modified Huffman technique the complexity is less but it is suitable for a small group and hence scalability is not supported.

In our proposed model we utilise the merits of Boolean minimization technique and Modified Huffman technique to achieve the security as well as better complexity in the MANET. In MANET one node (normally cluster head) acts as the master node which controls the other nodes in its vicinity. Here Cluster head acts as a group controller. Group controller is responsible for creating and maintaining the group key. Additionally an Unique ID for every user in the group is given based on the Boolean technique. But the length of this unique ID is not same for all the nodes. The length is based on the probability of leave of a member from the group. So, based on the probability of leave different subgroups are formed with different unique ID lengths and this can be achieved The member join/leave is handled as discussed using Boolean technique for the subgroups

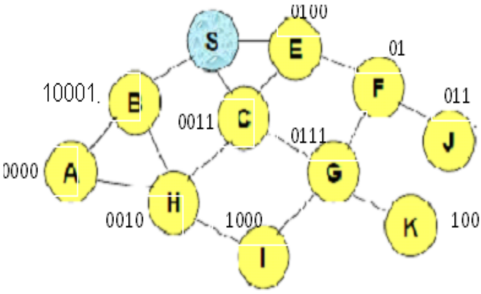


Figure 5 The figure 5 shows the various nodes in MANET with different length of UIDs and each node may represent one subgroup

K2

4. EXPERIMENTATION RESULTS AND COMPLEXITY ANALYSIS

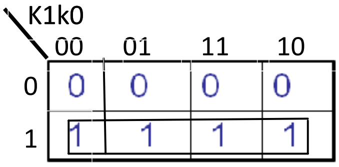
K2

4.1 Storage Complexity:

It includes the storage requirements of both group controller and users. The group controller has to store auxiliary keys of all the members and one group key. For example, the user size of 8, it has to store, k0k1k2, k0'k1'k2' and one group key which is equal to 2log28+1= 7. In general, the storage complexity of the server is 2log2n+1 where n is number of members. As for the storage requirement of the individual user is concerned, an user has to store all the auxiliary keys from the leaf node to the root and one group key. So, it is log2n+1.

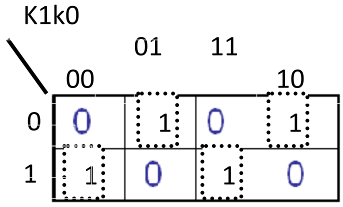
4.2 Communication and Computation Analysis:

Communication complexity is measured in terms of ' number of rekeying messages' sent by the group controller and computation complexity is measured in terms of number of encryptions needed by the group controller. Both complexities depend on the position of the existing members in the tree after the left out members. For example, in a user size of n=8, let us assume that four users leave the group with the available group members if the karnaugh map looks likes as shown in Figure 6 then minimizing the function, the adjacent ones may be combined which yield only one variable. This is referred as 'Best case'. Suppose if the four user's leaving position gives karnaugh map shown in figure 7, the minimization function becomes k2k1k0'+k2'k1'k0+k2k1k0+k2'k1k0', because the available ones cannot be combined. This is referred as the 'Worst case'. The experiments were simulated for various number of users and the results are tabulated in Table2,Table3,Table4 and Table5.



K1k0	00	01	11	10
0	0	0	0	0
1	1	1	1	1

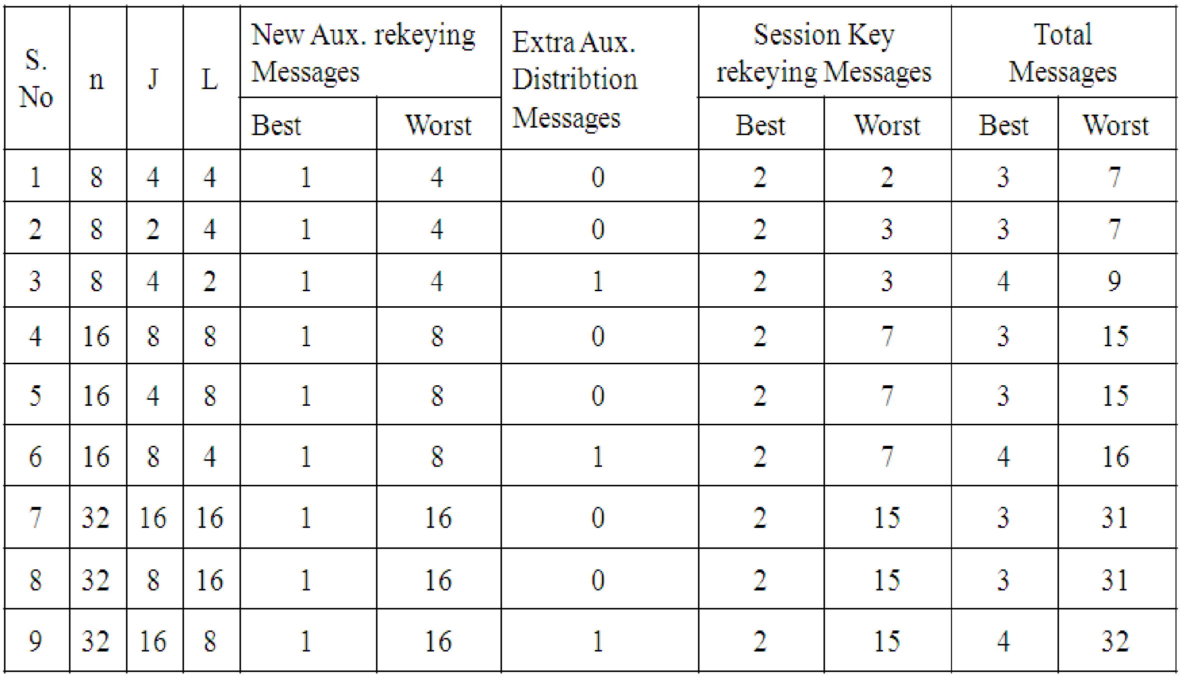
Figure 6. Minimization of member function in best case



K1k0	00	01	11	10
0	0	1	0	1
1	1	0	1	0

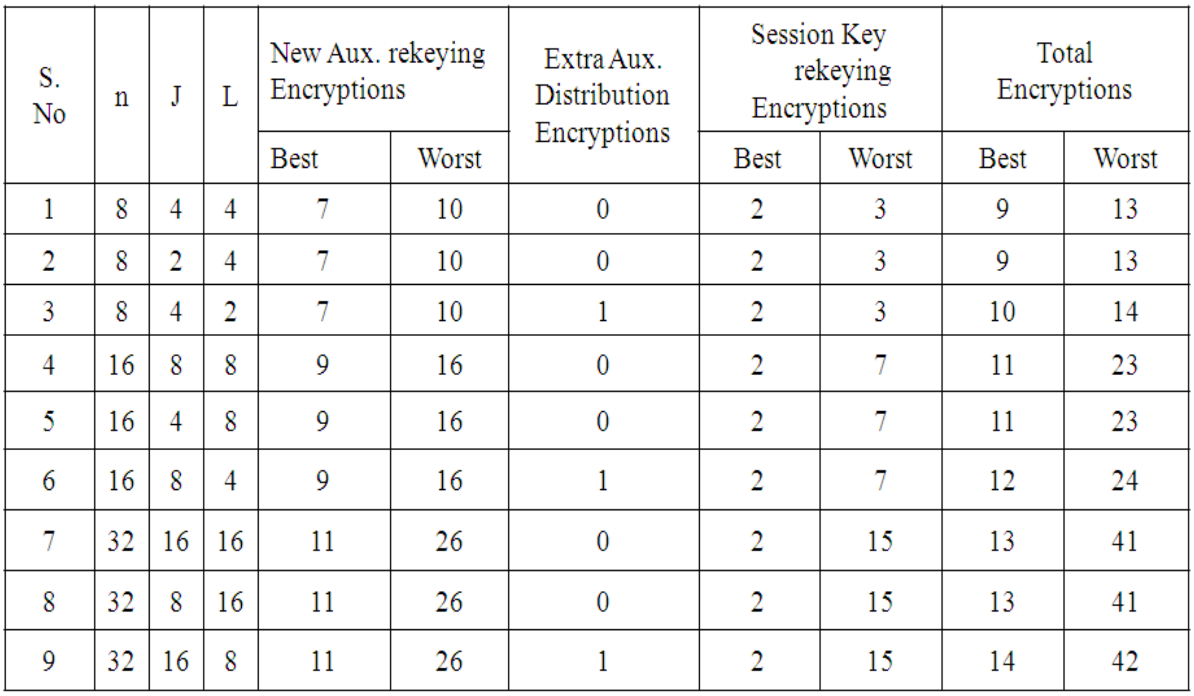
Figure 7 Minimization of member function in worst case

Table 2. Communication complexity of user size 8, 16 and 32



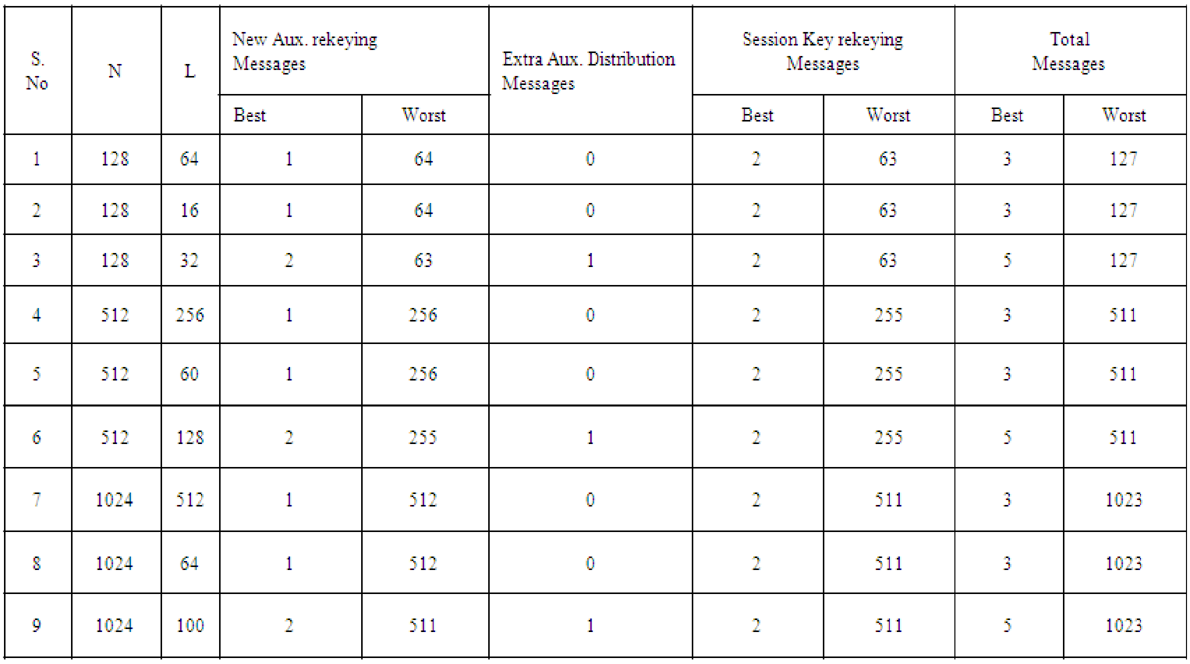
S. No	n	J	L	New Aux. rekeying Messages		Extra Aux. Distribtion Messages	Session Key rekeying Messages		Total Messages	
				Best	Worst		Best	Worst	Best	Worst
1	8	4	4	1	4	0	2	2	3	7
2	8	2	4	1	4	0	2	3	3	7
3	8	4	2	1	4	1	2	3	4	9
4	16	8	8	1	8	0	2	7	3	15
5	16	4	8	1	8	0	2	7	3	15
6	16	8	4	1	8	1	2	7	4	16
7	32	16	16	1	16	0	2	15	3	31
8	32	8	16	1	16	0	2	15	3	31
9	32	16	8	1	16	1	2	15	4	32

Table 3. Computation complexity of user size 8, 16 and 32



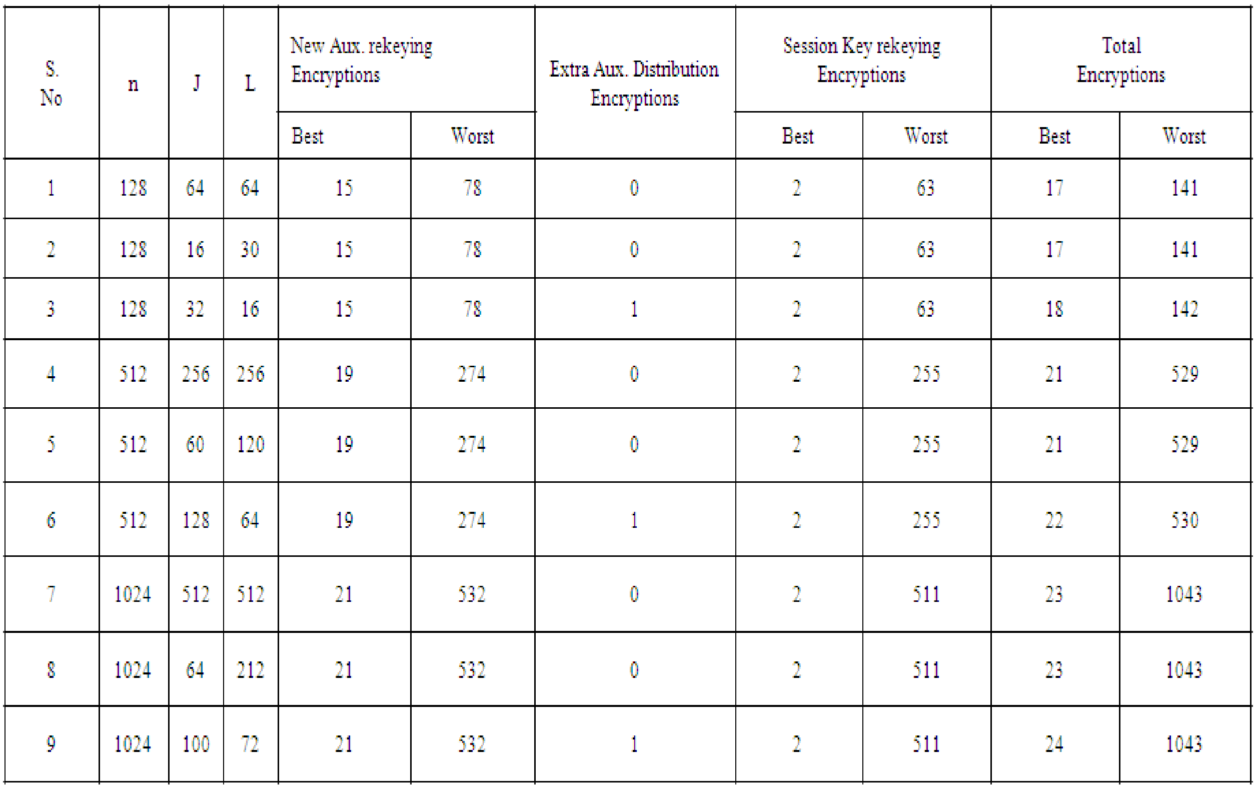
S. No	n	J	L	New Aux. rekeying Encryptions		Extra Aux. Distribution Encryptions	Session Key rekeying Encryptions		Total Encryptions	
				Best	Worst		Best	Worst	Best	Worst
1	8	4	4	7	10	0	2	3	9	13
2	8	2	4	7	10	0	2	3	9	13
3	8	4	2	7	10	1	2	3	10	14
4	16	8	8	9	16	0	2	7	11	23
5	16	4	8	9	16	0	2	7	11	23
6	16	8	4	9	16	1	2	7	12	24
7	32	16	16	11	26	0	2	15	13	41
8	32	8	16	11	26	0	2	15	13	41
9	32	16	8	11	26	1	2	15	14	42

Table 4. Communication complexity of user size 128, 512 and 1024



S. No	N	L	New Aux. rekeying Messages		Extra Aux. Distribution Messages	Session Key rekeying Messages		Total Messages	
			Best	Worst		Best	Worst	Best	Worst
1	128	64	1	64	0	2	63	3	127
2	128	16	1	64	0	2	63	3	127
3	128	32	2	63	1	2	63	5	127
4	512	256	1	256	0	2	255	3	511
5	512	60	1	256	0	2	255	3	511
6	512	128	2	255	1	2	255	5	511
7	1024	512	1	512	0	2	511	3	1023
8	1024	64	1	512	0	2	511	3	1023
9	1024	100	2	511	1	2	511	5	1023

Table 5. Computation complexity of user size 128, 512 and 1024



S. No	n	J	L	New Aux. rekeying Encryptions		Extra Aux. Distribution Encryptions	Session Key rekeying Encryptions		Total Encryptions	
				Best	Worst		Best	Worst	Best	Worst
1	128	64	64	15	78	0	2	63	17	141
2	128	16	30	15	78	0	2	63	17	141
3	128	32	16	15	78	1	2	63	18	142
4	512	256	256	19	274	0	2	255	21	529
5	512	60	120	19	274	0	2	255	21	529
6	512	128	64	19	274	1	2	255	22	530
7	1024	512	512	21	532	0	2	511	23	1043
8	1024	64	212	21	532	0	2	511	23	1043
9	1024	100	72	21	532	1	2	511	24	1043

5. CONCLUSION

Recent advancement in MANET have paved the way for group oriented application such as multicast communication which is the popular and efficient approach widely used for group communication. Due to the increased usage of group communication there is a heavy demand for security in multicasting. The security in multicasting imposes several problems and finding solutions to them become research challenges.

The most important feature of secure multicast is group dynamics i.e) the member of the multicast groups can join and leave the session at any time without intercepting the current session. In group dynamics the keys for the multicast group members are to be changed in order to maintain the forward and backward secrecy.

The main objective of this paper is to minimize the computational and communication costs involved in multicasting while changing the keys. An intelligence is embedded in to the group controller so that instead of assigning a constant length UID for the user, it assigns the UID based on the probability of leave. The results are encouraging and comparable with existing techniques.

REFERENCES

[1] Isabella Chang., Robert Engel, Dilip Kandlur., Dimitnos Pendarakis., Debanjan Saha,"Key Management for Secure Internet Multicast using Boolean Function Minimization Techniques," In proceedings of IEEE INFOCOM'99,New York,vol 2,pp.689-698(1999).

[2] Senthamil Ilango., Johnson Thomas,"Group Key Management utilizing Huffman and Petrick based approaches," Proceedings of the International Conference on Information Technology: Coding and Computing ITCC'04 (2004)

[3] V. Palanisamy., P. Annadurai, "Secure Geocast in ad hoc network using Multicasting Key Distribution Scheme," International Association of Computer Science and Information Technology - Spring Conference (2009)

[4] Mohit Choudhary., F'rashant Sharma., Dheeraj Sanghi, "Secure Multicast Model For Ad-Hoc Military Networks," (2004).

[5] Annadurai S., Purusothaman T, "Cost effective key management policies for pay programmes," CSI Communication (ISSN 0970-647X) (2005)

[6] Dondeti L, Mukherjee S. and Samal, "Scalable secure one –to-many group communication using dual encryption," Computer Communication,ACM,vol 23 pp.1681-1701 (1999)

[7] T.Srinivasan et.al 'A hybrid scalable group key management approach for large dynamic multicast networks' Proceedings of the sixth international conference on computer and information technology (2006)

[8] Mittra S Ilous, "A frame work for scalable secure Multicasting," in proc.ACM SIGCOMM'97,Cannes,France,vol.39,pp-277-288.

[9] Schneier B, "Applied Cryptography : Protocols, Algorithms and source code in C," 2nd Edition. Addision-Wesley (1995)

[10] Kim Y, Perrig A. and Tsudik, "Simple and fault –tolerant key agreement for dynamic collaborative groups," In proceedings of the 7th ACM Conference on Computer and Communications Security, vol.22, pp.235-244.(2000)

J.Lakshmana Perumal received B.E (EEE) from ACCET,Karaikudi and received M.E(CSE) from Government College of Technology, Coimbatore. He has 28 years of teaching experience in various Government Engineering Colleges and now he is professor and head of the department of Electrical and Electronics Engg., Government College of Engg., Salem-636 011, Tamil Nadu.

Dr.K.Thanushkodi received the BE in Electrical and Electronics Engineering from Madras University, Chennai. MSc (Engg) from Madras University, Chennai and PhD in Electrical and Electronics Engineering from Bharathiar University, Coimbatore in 1972, 1976 and 1991 respectively. His research interests lie in the area of Computer Modeling and Simulation, Computer Networking and Power System. He has published 26 technical papers in National and International Journals.

Prof.N.M.Saravanakumar received his M.E in Computer Science & Engineering and is now pursuing his Ph.D. in Secure Group Communication in Computer Science in Anna University of Technology, Coimbatore. Currently, he is working as a Senior Lecturer in CSE Dept, Bannari Amman Institute of Tech, Sathyamangalam, Erode, Tamil Nadu, India. He has 10 years of experience in teaching field. He has so far published 12 papers in various National Conferences and he has presented 2 papers in International Conferences held at various reputed engineering colleges. His interests include Network Security and Distributed Systems.

Prof K.Saravanan received a Masters Degree in ECE from Pondichery University, Pondichery.His research interests are of Mobile Adhoc Networks and Network Security.Currently, he is doing research at GCT,Coimbatore under Anna University of Technology,Coimbatore.

Mr. D.Vigneshwaran received the B.E. (EEE) from ACCET,Karaikudi and currently pursuing M.E at Government College of Technology, Coimbatote.

Dr.T.Purusothaman received M.E from GCT,Coimbatore and Ph.D. from Anna University,Chennai and he is currently working as a Associate Professor in CSE and IT department,Government College of Technology, Coimbatore.