Proceedings of the 2013 International Conference on
Pattern Recognition, Informatics and Mobile Engineering (PRIME) February 21-22

GAMANET: A Ganatic Algorithm approach for Hierarchical Group Key Management in Mobile Adhoc Network

K. Pushpalatha
Department of Computer Science
Anna University of Technology
Chennai, India
pushpalathakrishnan@gmail.com

Dr. M. Chitra
Department of Computer Science
Sona College of Technology
Salem, India
drmc@sonatech.ac.in

Abstract—**Mobile adhoc network (MANET) is a seamless integration of nodes that can be sender, recipient or relay and may unaware until they come in contact with each other in a decentralized network. Communication should takes place in a secure manner even with the changes on topology, bandwidth, network size, resources etc. The core aspect of establishing trust among the mobile nodes can do with the help of authentication check by exchanging keys. In this paper, we propose a genetic algorithm approach for hierarchical group key management scheme by simple rekeying technique on frequent scalable and high mobility nodes. The rekeying approach (in our scheme) distributes key with decentralize authority require single round and achieves consumption of energy during exchanging of rekeying messages with simplified number of bits. Few approaches such as OFT, CRTDH, GDH, and BH uses 3 rounds and DLKH, DOWF uses based on number of members in the network. The proposed approach (On our method) reduces energy consumption, computation, and communication compare to the existing scheme.**

Keywords- MANET; key-management; re-keying approach; genetic algorithm; Energy efficiency

I. INTRODUCTION

Mobile adhoc network is a seamless integration of nodes that can be sender, recipient or relay and may unaware until they come in contact with each other in a decentralized network. Communication should takes place in a secure manner even with the changes on topology, bandwidth, network size, resources etc. The nodes create network as a small group and stay connected to perform some specific task. Any node may join and leave the network at any point of time leads to dynamic changes in the network that tends to exchange information through multi-hop neighbor. To enable authenticity and avoid vulnerability, on membership requires offline trusted authority during system initialization for secure key distribution.

Group key needs to be established among multiple nodes to ensure secure exchange of information in the group which has common secret. The random movement of nodes needs updating of its membership thereby updating group key. The creation of group key and rekeying with no central trusted entity can be done periodically based on either 1-party Diffie-Hellman (DH) protocol, 2-party Diffie-Hellman or n-party Diffie-Hellman protocol before exchanging it between two unknown nodes or after every time the membership changes. This process is the backbone of secure group communication and gets more attention in current researches.

The efficient group key update to ensure backward and forward secrecy becomes active research area recently on collaborative and group-oriented MANET applications. The frequent topology changes due to erotic mobility and low resources availability creates serious issues in large and dynamic group key management on MANET over security and scalability.

We propose a genetic algorithm approach for self organized simple computational group key without central authority makes users themselves distributes keys and effects on energy consumption, less rekeying computation and communication cost over existing schemes. Basically the network layout classify clusters has 8 members and one acting as head, one cluster head out of 8 become group head for that group. Finally one among the group leader becomes network head for the entire network when single group key is not possible on frequent and erotic mobile nodes in scalable network.

The leader assigned special roles then other members during initialization phase, communication over inter group and inter cluster level. After the assignment of nodes any cluster member can begin the rekeying process thereby reducing extra load on the transmission power and memory resource of the leader. Group key helps the communication among cluster members. The gateway nodes supports inter cluster communication if the cluster heads situated far-off cluster. Similarly via group leaders the inter group communication performed. To ensure forward secrecy the rekeying messages created by member encrypt with public and private key pair before exchange.

The rest of the paper is organized as follows. Section II focuses on literature survey on relevant field. Section III presented proposed work. Section IV has given performance analysis and section V and VI focuses about results and conclusion, further enhancement.

978-1-4673-5845-3/13/$31.00©2013 IEEE

II. Related Work

Key management is a basic of all secure communication. The efficient, robust and secure key management is essential in most cryptosystem. Sharing secure information by common secret created by multiple members known as group key establishment. Secure group communication means that exchanging of secure information among members in a group which are inaccessible by the nodes out of network. The security ensured by encrypting the information with the group key established by all the participating members of that group, that leads decryption of message only to the group members designated to it. The typical classification of group key management protocol has four types such as centralized group key distribution (CGKD), decentralized group key management (DGKM), distributed/contributory group key agreement (CGKA) and distributed group key distribution (DGKD).

In CGKD, the group controller (GC) is a central entity is responsible for generating, distributing and updating the group key. The key tree scheme or logical key hierarchy (LKH) [1] is the popular one in CGKD scheme. In LKH root node specifies group participant or user in its tree structure that significantly reduces memory space and number of broadcasting messages over the root and leaf nodes. The user shares a pair wise key with group initiator as well as set of intermediate keys from leaf to the root. Similar to LKH another group key management approach is the one way function tree (OFT) is proposed in [2].

In [4] the extended version of Bohio and Miri [3] the authors present two variants of their former scheme to come up with group keys hidden to the TA: the primary scheme relies on group identity. Group public key QGRP-ID is to be generated by the TA supported any group identity or discretional string. The TA, using its master keys, then computes the initial group key D = s.QGRP_ID. Each node I will then receive the purpose D from the TA and can generate its personal key ki, a random secret, and figure the corresponding public key as Di-pub = ki.D. All such individual public keys ought to be offered from the TA. The collaborating nodes then get the general public key of each node from the TA. For the published key, parameter P1-brdcst = K1N.P is computed as within the basic theme with K1N being any random secret. The second scheme relies on individual identity. Theta can calculate the partial personal key of any node i as Di = s.Qid-i. Node i computes its private key as ki = H3(xi. Di), Where xi is a random secret chosen by node i. It computes public key as Di Di-pub = ki.P, and submits it to the TA element. The pair wise and broadcast keys are computed equally because the first scheme does.

Table of notations:

Symbols	***Meanings***
Z	Set of integers
Zn	Set of integers mod n
Fq	The finite field with q elements
Z^*_q	The multiplicative group of integers modulo prime number q. Z^*_q={a|1≤a≤q-}
E/F_p	Elliptic curve over F_p
G_1	Subgroup of the additive group of points of E/F_p
G_2	Subgroup of the multiplicative group of the finite field F^*_{p2}
$\hat{e} : G_1{*}G_1 \rightarrow G_2$	A bilinear map between two cyclic groups $G_{1,}G_2$
P	An arbitrary point in E/F_p
d_{ID}	Private key of ID
Q_{ID}	Public key of ID
S	Master secret key
P_{pub}	System public key
H(i)	A hash function. When multiple hash functions are used in a system, an integer i is used as subscript

The DGKM approach involves, dividing of small subgroups from larger groups. The subgroup controller in every subgroup is responsible for its key management. IOLUS [5] for scalable and secure multicasting is the first DGKM scheme. The CGKA scheme exchange information by absence of GC by all members contributes the group key in key management. The CGKA scheme typically includes binary tree based [6] n-party Diffie-Hellman key agreement [7, 8]. The tree based group keys, ensures secure and fully distributed protocol proposed in [6], but the main focus is to combine the contributory feature of DH with the efficiency of the tree structure.

The DGKD scheme in [9] introduces the concept of sponsors and co-distributors just by eliminating the need for a trusted central authority. All group members have equal responsibility and trust with same capacity could be a potential sponsor of a co-distributor or other members. The rekeying process initiates the sponsor of members, whenever it joins or leaves the group. The sponsor initiates the secure distribution of keys to the co-distributors once necessary keys generated. The co-distributors distribute parallel from corresponding key to corresponding members. The hierarchy and cluster based schemes [8, 10] are other classifications of key management technique and SGC has contributory group key agreement as the most appropriate technique in this kind of environment.

TABLE I. COMPARING GROUP KEY MANAGEMENT TYPES

<table>
<tr><th></th><th colspan="2">Centralized</th><th>Collaborative</th></tr>
<tr><td>Key management type</td><td colspan="2">Key distribution by key center</td><td>Key agreement by member’s contribution</td></tr>
<tr><td rowspan="2">Computation costs</td><td>Key center</td><td>Member</td><td rowspan="2">Large (similar complexity)</td></tr>
<tr><td>Large</td><td>Small</td></tr>
<tr><td>Features</td><td colspan="2">Single point of failure of key centre</td><td>Multiple communication rounds</td></tr>
<tr><td>Examples</td><td colspan="2">Key graph [14], OFT [11]</td><td>GDH [15], TGDH [13], STR[12]</td></tr>
</table>

The group members in MANET compute the group key in a distributed fashion have proposed in simple and efficient group key (SEGK) management scheme [16]. A sequential multisource model BALADE [17] optimizes bandwidth and energy consumption, with localization and nodes mobility as its parameters. Many of the approaches involve lot of exponentiations and complex operations increases computational burden and cost which is not suitable in erotic mobility of nodes. The group Diffie Hellman (GDH) calculates intermediate values in distributive fashion after group agrees on a pair of primes since all members must contribute to generate the group key therefore the size of the messages increases as the sequence is reaching the last members, as setup time linear and more intermediate values are necessary hence it is not suitable for large networks

Another approach, distributed one-way function tree (D-OWT) in [18] and its member generates own key and distributes blinded version of this key to its sibling using logical key hierarchy (LKH). Diffie Hellman logical key hierarchy (DHLKH) approach minimizes number of keys held by group member by generating keys, in the upper level using a one-way function (OWF).

The Chinese Remainder Theorem Diffie Hellman (CRTDH) is impractical in terms of efficiency, possessing the same least common multiple (LCM) by agreeing two large primes by computing group key as XOR operation of certain values. The evaluation of LCM was eliminated in modified CRTDH, obtains large value derived from CRT broadcast by one of its members. Due to no properly from obtain this message by leaving member. In addition table 1 shows the other types of key management techniques.

This paper proposes a distributed approach using genetic algorithm in which members contribute to the generation of group key by sending hash of a random number by sending hash of a random number during initialization phase regeneration within the cluster. During rekeying phase the regeneration of group key obtained messages from one of its member whenever membership update needed. The communication among cluster heads using group leader and transmitted securely to the other cluster heads, wherein the network head generates the key and passes on among the group leaders by applying same procedure. Symmetric key is used for communication between the cluster members and asymmetric key cryptography for distributing the rekeying messages to the members of the cluster.

III. Prepare Your Paper Before Styling

The design of this protocol is based on these notions:

- Key management should not rely on secure routing.
- Secure keys should be available before a routing protocol starts working.
- Secure routing starts from secure broadcasting.
- To prevent routing attacks, a routing protocol must encrypt and authenticate every message and packet, not only end-to-end, but also hop-by-hop.
- Some routing protocols have security or efficiency weaknesses.

With the secret system parameters, the nodes communicate with each other securely and set up routing table. The only way of communication before routing setup is broadcasting. The scheme utilizes system parameters of IBC to derive node-specific broadcast keys. These node specific broadcast keys are used to broadcast routing messages to all neighbors of a node or all other nodes in the network.

The node-specific broadcast keys, or in other words, 1-to-m keys, are essential for secure routing: pair-wise, or 1-to-1, keys cannot be used in routing protocols, because there is no routing between any two nodes; group-wise, or m-to-m, keys are not secure enough, because there is no authentication or non-repudiation, and is especially vulnerable to compromise because one compromised key reveals all encrypted messages for the whole group.

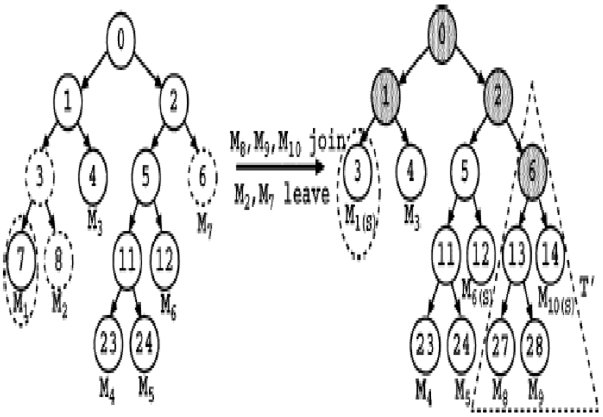

Figure 1. Example of queue merge phase

A. Pseudo-code of the queue sub tree phase

Queue-sub tree (T'):

```
if (a new member joins){
if(T'==NULL)/*no new member in T'*/
  create a new tree T' with the only new member;
else{/*there are new members in T'*/
  find the insertion node;
  add the new member to T';
  elect the rightmost member under the sub tree rooted at the
  sibling of the joining node to be the sponsor;
if(sponsor)/* sponsor's responsibility*/
  re-key renewed nodes and broadcast new
  Blinded keys;
}
}
```

Thus a more effective algorithm Queue-batch algorithm is proposed to develop. It reduces the rekeying load by pre-processing the joining members during the idle re-keying interval. The Queue-batch algorithm is divided into two phases, namely the Queue-sub tree phase and the Queue-merge phase. The first phase occurs whenever a new member joins the communication group during the re-keying interval. In this case, append this new member in a temporary key tree.

B. Pseudo-code of the Queue merge phase

```
                Queue-merge (T, T', $M^l$, L):
if (L==0){/* There is no leave*/
    add T' to either the shallowest node (Which need to be the leaf node) of T such that the merge will not increase the resulting tree height, or the root node of T if the merge to any location will increase the resulting tree height;
}
else {/* there are leaves*/
    add T' to the highest leave position of the key tree T;
    remove remaining L-1 leaving leaf nodes and promote their siblings;
}
    elect members to be sponsors if they are the rightmost members of the sub tree model rooted at the sibling nodes of the departed leaf nodes in T, or they are the rightmost member of T';
if(sponsor)/*sponsor's responsibility*/
    re-key renewed nodes and broadcast new blinded keys;
```

The second phase occurs at the beginning of every re-keying interval and we merge the temporary tree (which contains all newly joining members) to the existing key tree. To illustrate consider Fig. 1 where M8, M3 ,M10 join and M2, M7 leave and a temporary tree is merged with the existing tree.

C. Analysis of the queue-batch algorithm

The main idea of the Queue-batch algorithm exploits the idle rekeying interval to pre-process some re-keying operations. When we compare its performance with the Rebuild or Batch algorithms, we only need to consider the re-keying operations occurring at the beginning of every re-keying interval. When J = 0, Queue-batch is equivalent to Batch in the pure leave scenario. For J>0, the number of renewed nodes in Queue-batch during the Queue-merge phase is equivalent to that of Batch when J = 1.

IV. Performance Analysis

To reflect the latency of generating the latest group key for data confidentiality, we evaluate the performance of the interval-based algorithms using simulation-based experiments. Our simulation results show the Queue batch algorithm performs best among the others. The analysis of the two proposed algorithm are based on two performance measures i.e., number of exponentiation operations and the number of renewed nodes. The number of exponentiation operation gives a measure of the computation load in terms of node density to communication group's packets drop (Fig. 2).

TABLE II. Performance Analysis Of Group Key Management Protocols

			Member	Controller
OFT	Number of keys		$2\log_2 n$	2n-1
	Computation costs		$O(\log n)$	$O(\log n)$
	Messages sent on join/leave		$2\log_2 n$	
	Communication			Computation
GDH		Rounds	Messages	Exponentiation
	Join	4	n+3	n+3
	Leave	1	1	n-1
	Merge	m+3	n+2m+1	n+2m+1
	Partition	1	1	n-p
STR	Join	2	3	4
	Leave	1	1	(3n/2)+2
	Merge	2	3	3m+1
	Partition	1	1	(3n/2)+2
TGDH	Join	2	3	3h/2
	Leave	1	1	3h/2
	Merge	2	3	3h/2
	Partition	h	2h	3h

n – the number of members in the group
h – the height of the key tree
m – the number of merging groups
p – the number of members partitioned from a group of n members

V. Result

The existing key tree is totally balanced before the interval-based re-keying event. Each existing member has probability leave likelihood. The computation of the blinded cluster key of the basis node is counted within the blinded key computations. With this assumption, the amount of unsighted key computations merely equals the amount of revived nodes, only if the blinded key of every renewed node is broadcast just one occasion.

The figure above provides an inference of the batch re-keying to queue batch re-keying algorithm performance comparison in terms of re-keying interval to the no. of renewed nodes. It shows reduced no of renewed nodes for queue batch re-keying algorithm compared to that of batch re-keying model in various re-keying intervals.

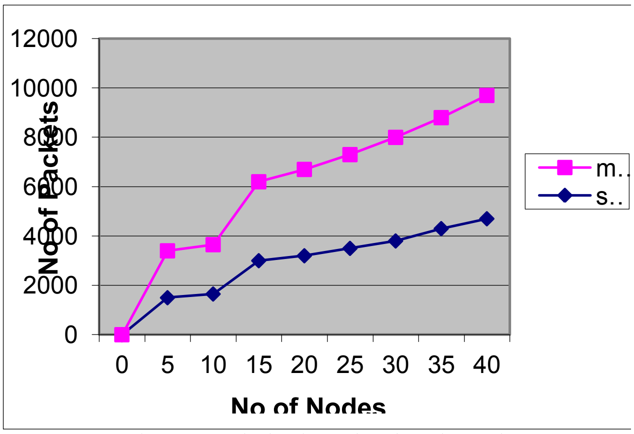

Figure 2. Node density and packets dropped

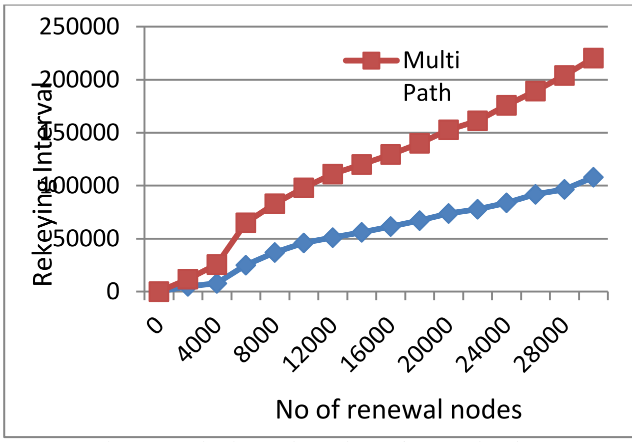

Figure 3. Rekeying nodes and no. of renewed nodes

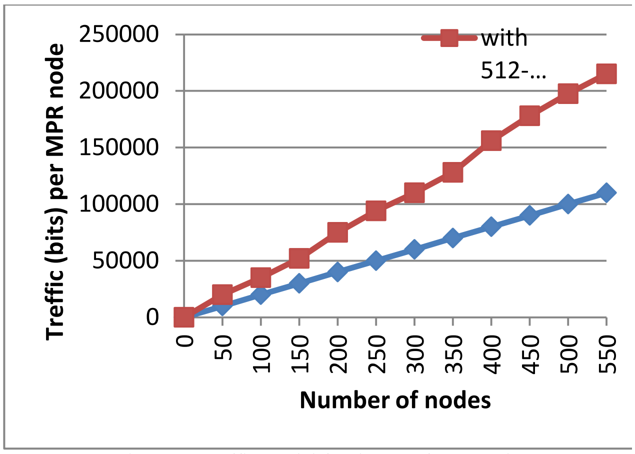

Figure 4. Traffic model for the KM framework

The number of renewed node is said to be renewed if it is a non leaf node and its associated keys are renewed.

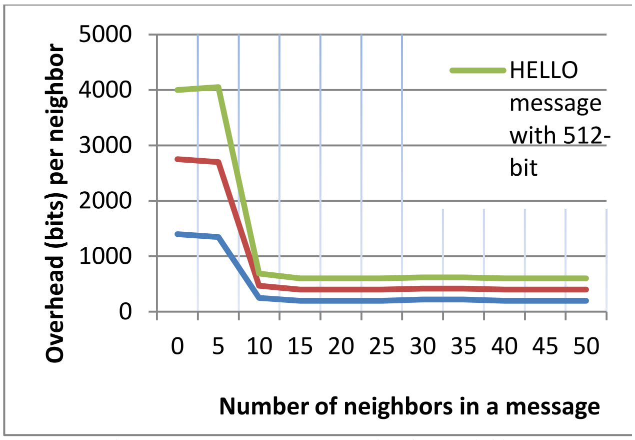

Figure 5. HELLO message overhead per neighbor

These metric measures the communication cost since the new blinded keys of the renewed nodes have to be broadcast to the whole group (Fig. 3). Traffic model for the KM framework and the HELLO message overhead per neighbor also shown in Fig. 4 and Fig. 5.

VI. Conclusion

Our simulation results shows that the Queue-batch algorithm can significantly reduce both computation and communication costs when there is highly frequent membership events. The proposal also addresses both authentication and implementation for the interval based key agreement algorithms. The proposed model of this study provides a distributed collaborative key agreement protocols for dynamic peer groups. The key agreement setting is performed in which there is no centralized key server to maintain or distribute the group key. To reduce the rekeying complexity, we propose to use an interval-based approach to carry out re-keying for multiple join and leave requests at the same time, with a tradeoff between security and performance.

VII. Using the Template

During this paper, tend to propose a KM and rekeying mechanism as the first part of our work later we will extend with SR as integrated scheme that addresses KM–SR interdependency cycle drawback. By exploitation identity based cryptography (IBC), this scheme provides safety features together with confidentiality, integrity, authentication, freshness, and non-repudiation. Our focuses on providing economical multicast communication on such networks.

References

[1] Wallner, D.M., Harder, E.J. and Agee, R.C., “Key management for multicast: issues and architectures,” Internet Draft, draft-wallner-key-arch-01.txt, 1998.

[2] Sherman, A.T. and McGrew, D.A., “Key establishment in large dynamic groups using one-way function trees,” IEEE Transactions on Software Engineering, vol. 29, no. 5, 2003, pp.444– 458.

[3] M.J. Bohio, A. Miri, “An authenticated broadcasting scheme for wireless ad hoc network,” in: Proc. CNSR 2004, IEEE Computer Society, 2004, pp. 69–74.

[4] M.J. Bohio, A. Miri, “Efficient identity-based security schemes for ad hoc network routing protocols, “ J. Ad Hoc Netw. 2 (3), 2004, pp. 309–317.

[5] S. Mittra. Iolus, “A framework for scalable secure multicasting,” Journal of Computer Communication Reviews, 27(4):277–288, 1997.

[6] Y. Kim, A. Perrig, and G. Tsudik., “Tree-based group key agreement. ACM Transactions on Information Systems Security,” 7(1):60–96, Feb. 2004.

[7] Y. Amir, Y. Kim, C. Nita-Rotaru, J. L. Schultz, J. Stan, and G. Tsudik, “Secure group communication using robust contributory key agreement,” IEEE Trans. Parallel and Distributed Systems, 15(5):468–480, 2004.

[8] [8] M. Burmester and Y. Desmedt,. “A secure and efficient conference key distribution system” In Advances in Cryptology - EUROCRYPT, 1994.

[9] P. Adusumilli, X. Zou, and B. Ramamurthy, “DGKD: Distributed group key distribution with authentication capability,” Proceedings of 2005 IEEEWorkshop on Information Assurance and Security, West Point, NY, USA, pp. 476–481, June 2005.

[10] J.-H. Huang and S. Mishra, “Mykil: a highly scalable key distribution protocol for large group multicast,” IEEE Global Telecommunications Conference, (GLOBECOM), 3:1476– 1480, 2003.

[11] D. Balenson, D. McGrew, and A. Sherman, "Key management for large dynamic groups: one way function trees and amortized initialization," IETF Internet Draft: draft-balensongroupkeymgmt-oft-00.txt, 1999.

[12] Y. Kim, A. Perrig, and G. Tsudik, "Communication-efficient group key agreement," in Proceedings of IFIP-SEC 2001, 2001, pp. 229-244.

[13] Y.Kim, A. Perrig, and G. Tsudik, "Simple and fault-tolerant key agreement for dynamic collaborative groups," in Proceedings of 7th ACM Conference on Computer and Communications Security, 2000, pp. 235-244.

[14] C. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs," ACM SIGCOMM '98, 1998, pp. 68-79.

[15] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups," IEEE Transactions on Parallel and Distributed Systems, vol. 11, 2000, pp. 769-780.

[16] Bing Wu, Jie Wuand Yuhong Dong, "An efficient group key management scheme for mobile ad hoc networks," Int. J. Security and Networks, 2008.

[17] MS. Bouassida, I. Chrisment, and 0. Festor, "A Group Key Management in MANETs. In International Journal of Network Security," vol.6, no. 1, pp.67-79, Jan. 2008 .

[18] Dondeti L., Mukherjee S., and Samal A., "A distributed group key management scheme for secure many-to-many communication," Tech. Rep. PINTL-TR-207-99, Department of Computer Science, University of Maryland, 1999.

[19] Diffie, W. and M. Hellman, "New directions in cryptography," IEEE Trans. Inf. Theory, 22: 644- 654. DOI: 10.1109/TIT.1976.1055638, 1976.

[20] Kim, Y., A. Perrig and G. Tsudik, "Communication Efficient Group Key Agreement," Trusted information, Springer, New York, ISBN: 0792373898, pp: 229-244, 2001.

[21] Kim, Y., Y. Kim and G. Tsudik, "Tree-based group key agreement," ACM Trans. Inf. Syst. Sec., 7: 60-96. DOI: 10.1145/984334.984337, 2004.

[22] Li, X.S., Y. R. Yang, M.G. Gouda and S.S. Lam, "Batch rekeying for secure group communications," Proceeding 10th International Conference on World Wide Web, (WWW'01), ACM New York, NY, USA, pp: 525-534. DOI: 10.1145/371920.372153, 2001.

[23] Amir, Y., Y. Kim, C. Nita-Rotaru, J.L. Schultz and J. Stanton et al., "Secure group communication using robust contributory key agreement," IEEE Trans. Parallel Distributed. Syst., 15: 468-480. DOI: 10.1109/TPDS.2004.1278104, 2004.