Performance Comparison of Distributed Group Key Management Protocol based on Region based Group Key Management

Deepali Jain
M. Tech (IT) Student, USICT
Guru Gobind Singh Indraprastha University
deepali_08@yahoo.com

Umang
Institute of Technology & Science
Mohan Nagar, Ghaziabad
umangsingh@its.edu.in

ABSTRACT

A mobile ad hoc network is an autonomous system of mobile nodes connected by wireless links. The nodes are free to move about and organize themselves into a network. For many application of MANET we need to establish secure group communication between members of groups. Some of the protocols to generate group key have been surveyed, and then their comparison is given based on the fact that if we divide the region into small region than the control overhead can be minimized. Numerical analysis is given to prove the result.

Keywords

Mobile ad hoc networks, Group communication, Group key management, Hierarchical group key management

1. INTRODUCTION

Mobile Ad hoc Network (MANET) is also known as a mobile mesh network [1]. It is an autonomous system of mobile nodes connected by wireless links. The nodes are free to move about and organize themselves into a network.

Mobile ad hoc networks does not require any fixed infrastructure such as base stations, therefore, it is an attractive networking option for connecting mobile devices quickly and spontaneously. or Some of the application of MANET is military operations, searching and rescue in disaster recovery, visiting an exhibition hall, and firefighters operating in a building.

The common characteristic of the above applications is that mobile nodes can be organized in the unit of groups, which could be further partitioned into many subgroups or merged with other groups [2]. In Ad Hoc networks all members communicating through wireless channels are more insecure and susceptible to numerous attacks than wired networks because radio channels used for communication in MANET is broadcast in nature and is shared by all nodes in the network. Thus, an attempt to establish secure group communications (SGC) over networks faces various challenges in order to meet security requirements.

In order to provide secure group communication secret session key is shared between group members. Maintaining secret key among group members is known as Group Key Management (GKM). The group is first established by initial members. Then one or several prospective members join the group while some members leave the group. A large number of membership changes, referred to as a bulk membership change, require a specialized protocol design without degrading group performance. In some scenarios a group can be partitioned into smaller subgroups or merged into a bigger group. This can also be considered a bulk membership change, but the transitions among groups likely incur overheads. This dynamic membership aspect requires the GCS to re-key the session keys in order to preserve the key secrecy.

Different approaches to group key management (GKM) are divided into three main classes:

1.1 Centralized group key management protocols.

A single entity is employed for controlling the whole group, hence a group key management protocol seeks to minimize storage requirements, computational power on both client and server sides, and bandwidth utilization;

1.2 Decentralized architectures

The management of a large group is divided among subgroup managers, trying to minimize the problem of concentrating the work in a single place;

1.3 Distributed key management protocol

There is no explicit KDC, and the members themselves do the key generation. All members can perform access control and the generation of the key can be either contributory, meaning that all members contribute some information to generate the group key, or done by one of the members.

The rest of the paper is organized as follows. First, some of distributed group key agreement protocol is analyzed. We have taken only distributed key agreement for comparison as there is no single point of failure, and divide the work equally into all nodes.

Then there is discussion on region based group key management. Section 4 describes the parameters taken for numerical analysis. Section 5 discusses the result of numerical analysis. Finally section 6 concludes the paper.

2. SURVEY OF EXISTING DISTRIBUTED GROUP KEY AGREEMENT ` PROTOCOLS

Notation

n	number of participant in the protocol
M_i	ith group member
K_n	Group key shared among n members

2.1 Group Key Distribution: GDH.1

The protocol (GDH.l) [3] is quite simple and straight-forward. It consists of two stages: upflow and downflow. The purpose of the upflow stage is to collect contributions from all group members.

Every Mi take the contribution from the member lower in the list, append its contribution on it, and then forward the result to Mi+1.

When Mn receives this value, it appends its contribution on it, which is the intended group key. In the downflow stage each node factor out its component and generate the group key Kn from the message and forward the result to member lower in the list.

In summary, GDH.l has following characteristics:

Rounds	2(n - 1)
Messages	2(n - 1)
Combined message size	(n - l) n
Exponentiations per M_i	(i + 1) for i < n, n for M_n
Total exponentiations	$\frac{(n+3)n}{2}$ - 1

The main drawback of GDH.l is its relatively large number of rounds but it imposes no special communication requirements, i.e., no broadcasting or synchronization is necessary.

2.2 Group Key Distribution: GDH.2

In order to reduce the number of rounds in GDH.l author has modify the protocol. The upflow stage is still used to collect contributions from all group members as in the GDH.1 protocol.

In the second stage Mn broadcasts the intermediate value to all group members. Every Mi then factors out its contribution and generate group key Kn

GDH.2 has the following characteristics:

Rounds	n
Messages	n
Combined message size	(n - l) (n/2+2) -1
Exponentiations per M_i	(i + 1) for i < n, n for M_n
Total exponentiations	$\frac{(n+3)n}{2}$ - 1

In GDH.2, more so than in GDH.l, the last node plays a special role by having to broadcast the last round of intermediate values. The main advantage of GDH.2 is due to its low number of protocol rounds; n as opposed to almost twice as many in GDH.l.

2.3 Group Key Distribution: GDH.3

The protocol consists of four stages. In the first stage we collect contributions from all group members similar to the upflow stage in GDH.1. After processing the upflow message, Mn−1 add its contribution to the result and broadcasts this value in the second stage to all other participants except Mn.

In the third stage, every Mi factors out its own exponent and forwards the result to Mn.

In the final stage, Mn collects all inputs from all other participants, raises every one of them to the power of Nn and broadcasts the resulting n − 1 values to the rest of the group.

Every Mi receives this message and can easily generate the intended secret key Kn.

GDH.3 has two appealing features:

Constant message size

Constant (and small) number of exponentiations for each node (Except for last with n exponentiation required, where n is the number of nodes in key generation)

The GDH.3 protocol has the following characteristics:

Rounds	n+1
Messages	2n-1
Combined message size	3(n-1)
Exponentiations per M_i	4 for i<(n-1) 2 forM_{n-1}, n for M_n
Total exponentiations	5n-6

2.4 Burmester/Desmedt Protocol

Burmester and Desmedt present in [4] a much more efficient protocol. The main idea in BD is to distribute the computation among members, such that each member performs only three exponentiations. Their protocol is executed in only three rounds:

1. Each user M_i generates its random exponent N_i and broadcasts

$$z_i = \alpha^{N_i} \tag{1}$$

2. Every M_i computes and broadcasts

$$X_i = \left(z_{i+1}/z_{i-1}\right)^{N_i} \tag{2}$$

3. M_i can now compute the key

$$K_n = z_{i-1}^{nN_i} \,.\, X_i^{n-1} \,.\, X_{i+1}^{n-2} \,..X_{i-2} \bmod p \tag{3}$$

In summary, the BD protocol has the following characteristics:

Rounds	2
Messages	2 n
Combined message size	2 n
Exponentiations per M_i	n+1
Total exponentiations	(n+1) n

An important advantage of the BD protocol is its “cheap” exponentiations. While the number of exponentiations per Mi is still (n + l), in all but one the exponent is at most (n - 1). This makes for big savings in computation.

2.5 CRTDH

In order to establish the group key in CRTDH protocol [5] every Mi broadcast the DH public share to all the members in the group. Every Mi then compute the DH key shared with each of them. Find the Least Common Multiple (LCM) of all the DH keys calculated. Perform calculation on it based on the Chinese remainder theorem (CRT) and broadcast the result to all group members. All group members then generate Kn based on the received value.

Main advantage of CRTDH is that members independently but mutually generate the group key.

In summary, the CRTDH protocol has the following characteristics:

Rounds	2
Messages	2 n
Combined message size	2 n
Exponentiations per M_i	n
Total exponentiations	n^2

An important advantage of the CRTDH protocol is its “cheap” exponentiations. In CRTDH there is no need for serialization and all nodes perform equal amount of work.

Table 1 Comparison of group key agreement protocols

	GDH.1	GDH.2	GDH.3	BD	CRTDH
Rounds	2(n-1)	n	n+1	2	2
Total messages	2(n-1)	n	2n-1	2n	2n
Combined message size	n(n-1)	$\frac{(n+3)n}{2}$ -3	3(n-1)	2n	2n
Messages sent per M_i	2 1for M_1,M_2	1	2	2	2
Messages recieved per M_i	2 1for M_1,M_2	2 1for M_1,M_2	3 n for M_n	2(n-1)	2(n-1)
Exponentiations per M_i	i+1	i+1	4 2 for M_{n-1} n-1 for M_n	n+1	n
Total exponentiation	$\frac{(n+3)n}{2}$ -1	$\frac{(n+3)n}{2}$ -1	5n-6	(n+1)n	n^2
Serialization	Y	Y	Y	Y	N
DH key	Y	Y	Y	N	Y
Symmetry	N	N	N	Y	Y

2.4 Comparison of Group Key Agreement Protocol

All group key distribution protocols discussed above are summarized and compared in Table 1. BD and CRTDH are markedly superior to the others with respect to exponentiation operations since almost all operations involve relatively small exponents. From Table 1 it is clear that, with respect to time (i.e., number of rounds), the BD and CRTDH protocol is well ahead of the rest. It requires only two rounds of simultaneous broadcasts as opposed to linear (in terms of number of rounds) in the other protocols.

On the other hand (n-1) simultaneous uncast in GDH.3 result in significantly less load as compared with n simultaneous broadcasts in BD and CRTDH.

Another important measure of protocol efficiency is the number of messages received and sent by each participant. It is well-known that sending or receiving a message involves going through the entire protocol stack - a non negligible task in terms of both time and resource consumption. Moreover, it is impossible in most (non-specialized) network architectures for a node to receive multiple messages simultaneously. This consideration is especially applicable to both BD and CRTDH protocols, i.e., regardless of whether all nodes can broadcast simultaneously, a given node cannot receive (n - 1) incoming messages all at once. Table 5 clearly illustrates that GDH.2 involves the least overhead with respect to the communication infrastructure: as part of the protocol each node sends a single message and receives only two (except Mi and Mn, which receive one message).

Now we consider the issue of protocol symmetry. BD and CRTDH offer symmetric operation. This is partly due to their synchronous nature. (An asynchronous protocol cannot be symmetric; someone has to initiate it.) All three GDH protocols are, to certain extent, asymmetric. GDH.1/2 are both communication-asymmetric. GDH.l requires Mi to initiate the upflow, and Mn, - the downflow, stage. GDH.2 is similar in that it requires Mn to perform the final broadcast. GDH.3 is not only communication- but also computation symmetric. The former is because Ml and Mn-1 are required to initiate sta.es 1 and 2, respectively. Computational asymmetry is due to the special role of Mn who has to perform computations different from those of other participants. (Note that Mn performs n-1 exponentiations in stage 4; however, it does not compute an

inverse of Nn

Finally, with regard to serialization of members, only CRTDH does not require member serialization. All other protocols need serialization of members.

3. REGION-BASED GROUP KEY MANAGEMENT PROTOCOL

In hierarchical group key management the operational area is break into regions based on decentralized control in order to reduce the group key management overhead and to make the protocol scalable to a large number of nodes in a group [6].

In this approach, every region has a regional leader, which communicates with other regional leader for key generation. In each region there is a regional key. Regional leader of each region share a leader key and generate group key for communication between members of different region. Keys need to be restructured if groups partitioning occur or group merge occur.

Other hierarchical group key management protocols proposed in the literature is:

Hardjono et al. [7] and Zhang et al. [8] presented IGKMP that divides a group into several subgroups to enhance scalability.

Rafaeli et al. [9] proposed HYDRA that divides a group into a number of TTL-scoped regions for flexible and efficient group key management to support secure multicasting.

Dondeti et al. [10] proposed DEP for secure multicasting based on a hierarchical subgrouping architecture for scalability.

Similarly, Iolus [11] is a framework that divides a group into smaller subgroups each with multiple subgroup controllers.

4. SIMULATION MODEL

We have used mathematical model presented in [6] to show that if we divide the total area into number of regions than we can reduce the group key management overhead. And there exists an optimal region size that minimizes the overall communication cost.

All five group key management protocol described in section 2 is applied to the mathematical model and then their comparison is given.

For simulation, radius of simulation area is taken as 1 km, node density is taken as 100 nodes/ km2, mobility rate per node is taken as 1/60*60, and wireless per hop radio range is taken as 250 meters. Area of region is π km2 as circular area has been chosen for calculation.

Matrix taken for numerical analysis is number of hop bits per sec i.e. total cost).

Results are evaluated for region size 1, 7, 19, 37, 61, 91 and 127.

5. RESULTS

Fig. 1 shows region sizes vs. group key management protocol

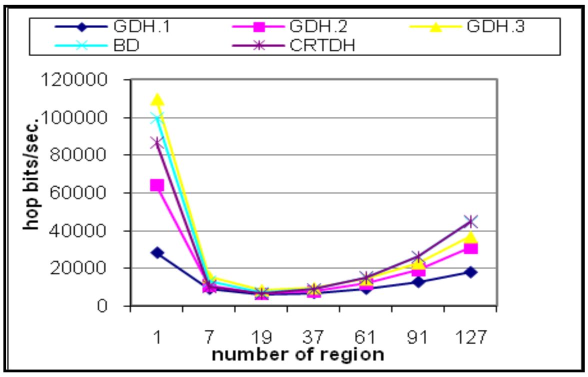


Fig. 1: Overall cost vs. no. of region as a function of group key management protocol

In the Fig., we see that as the number of region increases, total cost decreases until it reaches the optimal point at number of region 19 that would minimize total cost, after which total cost increases again beyond that point.

Note that higher number of regions indicates that there are fewer members in the region. The reason that an optimal number of region exists is that as number of region increases, the inter regional overhead increases (i.e. updating and rekeying cost at leader level), while the intra-regional (i.e. updating and rekeying cost at a regional level) overhead decreases. Initially, the total communication cost decreases as the number of regions increases because of the decreasing intra-regional overhead while it increases again after the optimal region size reaches because of the increasing inter-regional overhead.

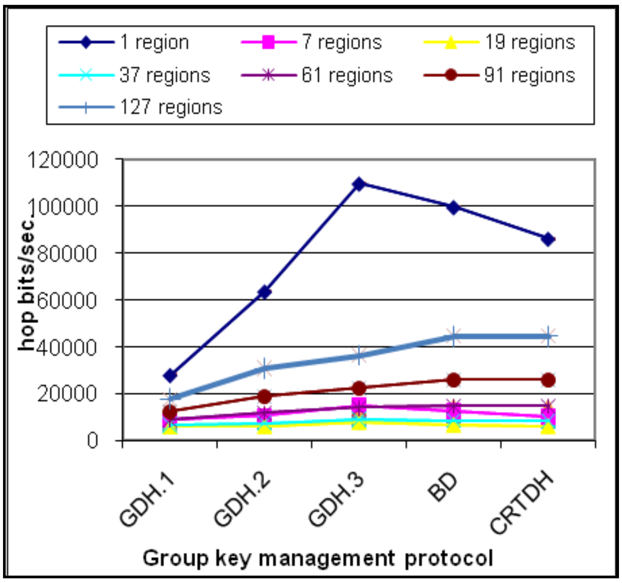


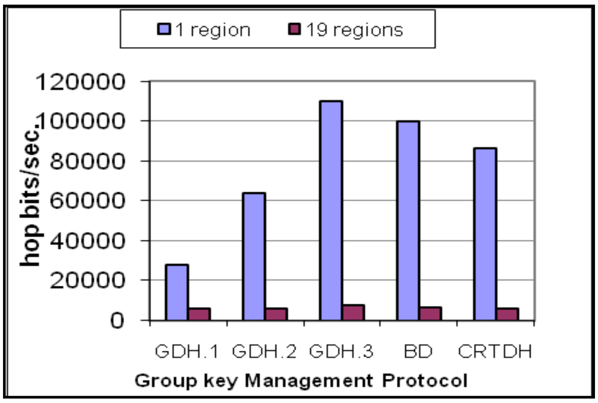
Fig. 2: Overall cost vs. group key management protocol as a function of no. of region

Fig. 2 shows another view of the Fig. 1. It shows that, when the number of region is 19 then total cost is minimum for all five protocols. For all other region size, cost is more.

Highest cost is achieved when the number of region is one i.e. the base case. Highest cost is achieved when the protocol is GDH.3 and number of region is one.

Least cost is obtained when the protocol is GDH.1 and the numbers of regions are taken as nineteen.

Fig. 3 shows the impact of number of region on the total cost between the 1 region systems vs. the optimum region size (19 regions). The network traffic generated under the optimal region size i.e. 19 is significantly lower than that under the no-region protocol.



.

Fig. 3: Overall cost in no region vs. in 19 regions as a function of group key management protocol

Fig. 4 shows the communication overhead of five protocols when the number of regions is 19(optimum region size). GDH.3 is more communication intensive as compared to other protocols, because it takes large number and messages, to generate a key. GDH.1 has less overhead in terms of number of messages exchanged to generate key, because it unicast to the member next in the list but as stated earlier in section 2 that it takes large number of rounds to generate the key.

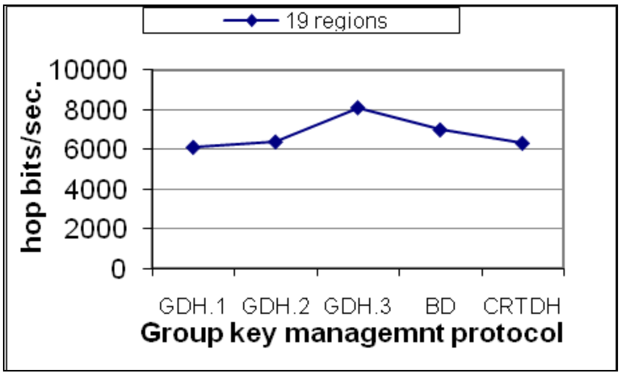


Fig. 4: Overall cost in 19 region as a function of group key management protocol

If we compare BD protocol with other protocol, it is less communication intensive than GDH.3 because it takes only two rounds of simultaneous broadcast but if we compare it with other three protocols it is more communication intensive. Although, CRTDH and BD both use two round of simultaneous broadcast to generate a key, but CRTDH use only one round of computation when a member leave the group, so the communication cost of BD becomes higher than that of CRTDH.

CRTDH and GDH.2 have almost same communication overhead. But CRTDH outperform GDH.2 as it does not require member serialization and take less number of rounds then GDH.2.

6. CONCLUSION

In this paper, we have surveyed group key management protocol, and their comparison is given in terms of number of rounds, message sent for key generation and computation power. Then we have used numerical method to show that if we divide our region into smaller region than the total communication cost can be minimized. And then comparison of these protocols is given. Numerical method chosen for the evaluation is based on the characteristic of MANET. It has been shown that for all protocols there exists an optimal region size when the cost of communication is minimum. In our opinion, CRTDH outperform all as it is not computation intensive, require no serialization of nodes, and less number of rounds for key generation and communication overhead is not very high.

7. REFRENCES

[1] Basagni, Conti, Giordano, Stojmenovic, "Mobile Ad Hoc Networking", IEEE Press, New Jersy, 2004.

[2] S. Rafaeli, D. Hutchison, A survey of key management for secure group communication, ACM Computing Surveys (CSUR) 35 (3) (2003) 309–329.

[3] M. Steiner, G. Tsudik, M.Waidner, Diffie–Hellman key distribution extended to group communication, in: Proc. 3rd ACM Conf. on Computer and Communications Security, January 1996, pp. 31–37.

[4] M. Burmester and Y. Desmedt. A Secure and Efficient Conference Key Distribution System. In I.B. Damgard, editor, Advances in Cryptology - EUROCRYPT '94, Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, 1994.

[5] R. K. Balachandran et al., "CRTDH: An Efficient Key Agreement Scheme for Secure Group Communications in Wireless Ad Hoc Networks," Proc. IEEE ICC '05, vol. 2, May 2005, pp. 1123–27.

[6] J.H. Cho, I.R. Chen, D.C. Wang, Performance optimization of region-based group key management in mobile ad-hoc networks, Performance Evaluation 65 (5) (2008) 319–344.

[7] T. Hardjono, B. Cain, I. Monga, Intra-domain group key management protocol, Internet Draft (1998).

[8] C. Zhang, B. DeCleene, J. Kurose, D. Towsley, Comparison of inter-area rekeying algorithms for secure wireless group communications, Performance Evaluation 49 (1–4) (2002) 1–20.

[9] S. Rafaeli, D. Hutchison, HYDRA: A decentralized group key management, in: Proc. 11th IEEE Int'l Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, June 2002, pp. 62–67.

[10] L. Dondeti, S. Mukherjee, A. Samal, Scalable secure one-to-many group communication using dual encryption, Computer Communications 23 (17) (2000) 1681–1701.

[11] S. Mittra, Iolus: A framework for scalable secure multicasting, in: Proc. ACM SIGCOMM'97, vol. 27, no. 4, Cannes France, September 1997.