Available online at www.sciencedirect.com
ScienceDirect
Performance Evaluation 65 (2008) 319–344
PERFORMANCE EVALUATION
An International Journal
www.elsevier.com/locate/peva

Performance optimization of region-based group key management in mobile ad hoc networks

Jin-Hee Cho[a], Ing-Ray Chen[a,*], Ding-Chau Wang[b]

[a] *Virginia Tech, Department of Computer Science, United States*
[b] *Southern Taiwan University of Technology, Department of Information Management, Taiwan*

Received 21 November 2006; received in revised form 27 February 2007; accepted 11 July 2007
Available online 26 July 2007

Abstract

We propose and analyse a scalable and efficient region-based group key management protocol for secure group communications in mobile ad hoc networks. For scalability and dynamic reconfigurability, we take a region-based approach by which group members are broken into region-based subgroups. Leaders in subgroups securely communicate with each other to agree on a group key in response to membership change and member mobility-induced events. We propose a novel approach to identify the optimal setting of the region-based key management protocol to maximize the performance of the system. We show that secrecy requirements for secure group communication are satisfied, and that there exists an optimal region size that minimizes the network traffic as a result of efficiently trading inter-regional vs. intra-regional group key management overheads. We compare the proposed region-based key management protocol with nonregion-based key management to demonstrate the effectiveness. Analytical results are validated by extensive simulation.

Keywords: Performance analysis; Mobile ad hoc networks; Group key management; Region-based key management; Secure group communication; Wireless networks

1. Introduction

A mobile ad hoc network (MANET) consists of a set of wireless mobile devices communicating with each other without an infrastructure. Since a MANET can be rapidly deployed, it has gained popularity in applications such as battlefields, disaster recovery efforts, emergency teams, group conferences, etc. In these applications, efficient and secure group communication is a prime concern. The challenges for realizing efficient and secure group communication in MANETs include constrained resources (i.e. CPU, memory, battery, bandwidth, etc.), unreliable communication and frequent changes of network topology induced by node mobility [30].

For efficiency, a commonly accepted approach for secure group communications is for group members to share a secret key (called a *group key* hereafter) [25]. When a member joins a group, the group key is rekeyed to ensure that the new member cannot decrypt previous messages, a security requirement known as *backward secrecy* [4]. When a member leaves the group, the group key is rekeyed to ensure that future communications cannot be decrypted by

* Corresponding author. Tel.: +1 703 538 8376; fax: +1 703 538 8348.
E-mail addresses: jicho@vt.edu (J.-H. Cho), irchen@vt.edu (I.-R. Chen), zh9@mail.stut.edu.tw (D.-C. Wang).

0166-5316/$ - see front matter
doi:10.1016/j.peva.2007.07.002

the leaving member, a security requirement known as *forward secrecy*. An algorithm that deals with the distribution, updating, and revocation of group keys is called a *group key management protocol*. Group key management to preserve secrecy is a very well-studied problem in wired networks and more recently in wireless communication. Many existing group key management protocols designed for wireless networks, however, cannot be applied to MANETs because they do not consider node mobility. Thus, the effect of mobility-induced events, such as group partitioning/merging and dynamic change of topology, on the security and performance of the system is not addressed.

In this paper, we propose and analyze a *region-based* group key management protocol for secure group communications in MANETs, extending our earlier work [41]. The protocol follows the hierarchical group key management concept to break the operational area 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. Unlike previous work, we especially deal with mobility-induced events such as group partitioning/merging and design security measures to deal with these events. In particular, we identify the optimal setting of the proposed region-based group key management protocol to maximize the performance of the system while preserving the backward/forward confidentiality requirements. We show that the region-based group key management protocol outperforms nonregion-based protocols over a wide range of parameter values that characterize the operational and workload conditions of a MANET.

The rest of this paper is organized as follows. Section 2 surveys related work and distinguishes our work from existing work. Section 3 describes our proposed region-based group key management protocol in detail. Section 4 gives the system model, security model, and attack model. Section 5 develops a performance model to evaluate performance characteristics of the region-based group key management protocol. Section 6 analyses the network traffic generated by our region-based key management protocol as a result of responding to events in the system, and identifies optimal region sizes under which the overall communication cost for group key management is minimized while satisfying forward/backward confidentiality requirements. Section 6 also provides a sensitivity analysis to evaluate the effect of main design parameters. Finally, Section 7 concludes the paper and outlines future work.

2. Related work

Group key management can be classified into *centralized*, *decentralized* and *distributed* [14,15,23]. A *centralized* scheme uses a key controller for key management tasks including key generation, assignment, distribution, revocation, etc. and is not suitable for MANETs. A *decentralized* scheme divides a group into subgroups typically hierarchically to spread out the workload of a central controller. A *distributed* scheme does not have a group key controller for group key management. Instead, a group key is generated in a contributory manner by all members in the system. The region-based group key management scheme developed in the paper is a hybrid of decentralized and distributed schemes. Similar to a decentralized scheme, it divides a group into region-based subgroups in a two-level hierarchy. Similar to a distributed scheme, however, there is no group key controller within each region, with all members contributing to key management.

Over the past few years, there have been hierarchical group key management protocols proposed in the literature. Hardjono et al. [13] and Zhang et al. [10] presented *IGKMP* that divides a group into several subgroups to enhance scalability. They proposed several rekeying algorithms that preserve secrecy properties as members move within the hierarchy. Within each subgroup, a set of key controllers is used for key management. This approach is suitable for wired networks since key controllers are stationary and wired communication to key controllers is reliable. However, it is not suitable for MANETs where nodes are mobile and wireless communication is often unreliable. Rafaeli et al. [18] proposed *HYDRA* that divides a group into a number of TTL-scoped regions for flexible and efficient group key management to support secure multicasting. *HYDRA*, however, is also based on the use of multiple group controllers in a region. Dondeti et al. [19] proposed *DEP* for secure multicasting based on a hierarchical subgrouping architecture for scalability. Again, *DEP* also uses multiple subgroup controllers. Similarly, *Iolus* [20] is a framework that divides a group into smaller subgroups each with multiple subgroup controllers. In [16], *HKT* was proposed to balance security and efficiency making use of a two-level hybrid key tree based on clusters. Cluster sizes are adjusted depending on the level of collusion resistance. We observe that these hierarchical group key management protocols are designed for wired networks and are not suitable for MANETs.

Most existing work on hierarchical group key management in MANETs considered hierarchical clustering for grouping nodes into clusters for scalability and efficiency [17,21,34–40]. Rhee et al. employed a two-layer hierarchical key management structure for secure group communication overseen by unmanned aerial vehicles (UAVs) [35]. They

considered the use of a stationary supernode as a cluster head. Bechler et al. proposed an efficient distributed key management for certification based on hierarchical clustering [36]. However, no optimal setting of their proposed protocol was identified to maximize system performance. Clustering algorithms in MANETs for energy conservation were proposed in [37,38,40] with the effort mainly focused on the effect of clustering on energy consumption assuming a predetermined cluster size. Lazos et al. [39] considered a hierarchical key management structure for energy-aware secure multicast group communication in MANETs based on geographical routing. Their work assumed a fixed cluster size without identifying the optimal cluster size. Further, events that could happen in secure group communications such as group join/leave, group partition/merge, etc. were not considered. Unlike previous work, [17,21,34] identified the optimal cluster size or the range of it in hierarchical clustering key management that would minimize the cost of clustering and rekeying. In particular, [34] proposed a hierarchical group key management scheme under which nodes are grouped into multiple clusters and keys are distributed such that the intra-cluster communication is secured based on symmetrical cryptography (i.e. using GDH.2) and the inter-cluster communication is secured based on asymmetrical cryptography. They identified an optimal cluster size to tradeoff low communication cost for symmetrical systems within a cluster vs. high computation cost for asymmetrical systems among cluster heads. However, no analysis was performed on the effect of group partitioning/merging and group member disconnect/reconnect events that can possibly occur in MANETs.

We assume that nodes are equipped with GPS and thus nodes are capable of knowing its geographical location. Instead of executing a clustering algorithm which wastes energy, nodes self-organize and group themselves into region-based subgroups. The region-based group key management scheme developed in this paper for MANETs derives from *IGKMP* [10] for decentralized group key management for scalability and efficiency. Unlike *IGKMP*, however, we adopt distributed key management within each region for robustness such that there is no single node acting as a key controller. All members within a region participate in key generation and management by following a distributed group key management scheme, thus removing a single point of failure. Our hybrid decentralized-distributed region-based group key management scheme encompasses efficiency, scalability, and robustness without relying on central entities for key management. All nodes are homogeneous and no supernodes are required. A major contribution of our paper is that we determine the optimal region size to be used to minimize network traffic due to group key management activities, in response to group join/leave, mobility, and partitioning/merging events.

A key design concept of our region-based group key management scheme is that a distributed scheme be used for key generation and management within a region to achieve robustness. In the literature, there have been quite a few distributed group key management schemes proposed in recent years. Group Diffie–Hellman (*GDH*) [8] extends from the well-known two-party Diffie–Hellman (*DH*) key exchange protocol to allow a number of nodes to agree on a shared secret key without having a secure channel. In particular, *GDH*.2, *GDH*.3 [8] and *CLIQUES* in [22] improved upon *GDH*. Based on *GDH*, Amir et al. [1–3] discussed a robust contributory key agreement protocol resilient to group membership changes. Becker and Wille [24] proposed *Octopus* based on *DH* key exchange. In addition, *ING* [31] extends from *DH* key exchange; *BD* [32] aims to reduce communication overheads based on public keys. Very recently distributed key management schemes based on logical key trees, e.g. Logical Key Hierarchy (*LKH*) [29], have been proposed. Kim et al. [5,28] proposed tree-based group key management schemes, *STR* and *TGDH*, combining the benefits of *GDH* and *LKH*. Rodeh et al. [7] proposed a decentralized key management protocol using key graphs. They later also suggested a distributed *LKH* scheme (*DLKH*) for which no key controller is needed and the logical key hierarchy is constructed by group members [26]. Distributed One-Way Function Tree (*DOFT*), an extension of *OFT* [33], was proposed in [27] to allow a group member to initiate access control and key generation. Lee et al. [42] developed distributed collaborated group key agreement protocols based on periodic batch rekeying [6] for performance optimization of distributed secure group communication systems.

All the above distributed key management protocols incur high communication overheads as they have been designed to apply to the whole group. The region-based group key management scheme proposed in this paper allows any of these existing distributed key management schemes to apply at the subgroup level (i.e. at the intra-regional level) to achieve robustness without sacrificing efficiency. A key design in our region-based group key management scheme is to identify the *optimal* region size to minimize network traffic due to key management operations and mobility-induced events in MANETs. The optimal region size depends on the distributed key management scheme adopted. In this paper, we exemplify how the optimal region size can be determined with *GDH*.

In our definition a group is a mission-orientated group in which members must share information and communicate with each other securely and timely for mission execution. A group only consists of nodes that are connected (through

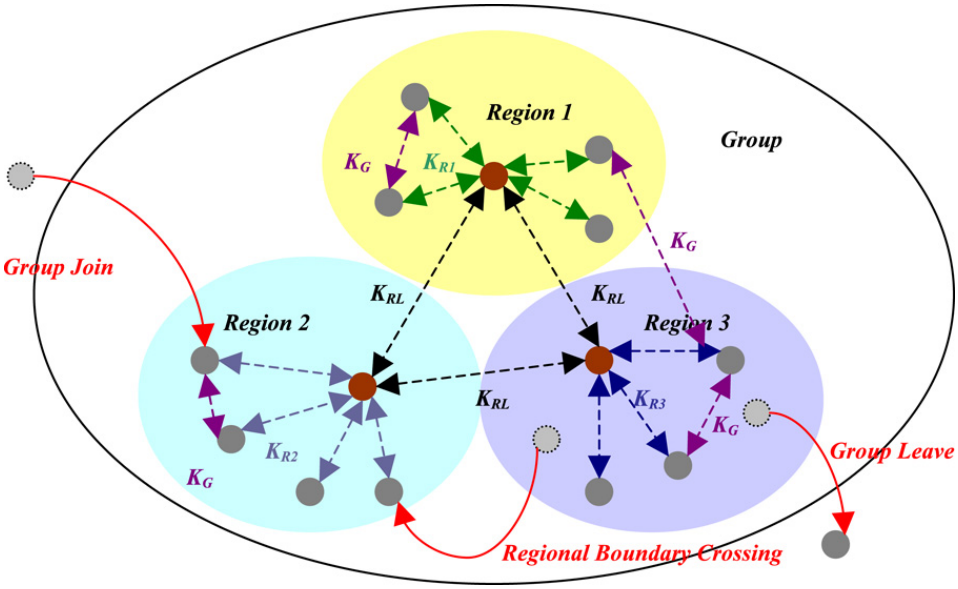


Fig. 1. Region-based group key management.

single-hop or multi-hop). Since nodes can move freely, network partitioning or merging may occur, thus possibly dividing a group into multiple groups or merging several groups into one group. At any time, however, a node may belong to only one group despite group partitioning/merging. This definition is different from prior work that considers the possibility of a node belonging to several groups, e.g. as in [43] that attempts to develop techniques such as packing the key trees of different groups into key bundles or key parcels for performance optimization.

3. Region-based group key management protocol for MANET

For scalability and efficiency, our region-based group key management scheme divides a group into region-based subgroups based on *decentralized* key management principles as illustrated in Fig. 1. Table 1 shows some of the symbols used to represent entities in the system. We assume that each group member is equipped with GPS and knows its location as it moves across regions. When a regional boundary is crossed, a member retains its group membership, but changes its subgroup "regional" membership. For secure group communications, all group members share a secret group key, K_G. On the other hand, for secure subgroup communications, all subgroup members in region i share a secret key, K_{Ri}. For robustness, a *distributed* key management scheme is used to generate and manage the shared secret key. In the discussion below, we assume that a contributory key agreement protocol such as *GDH* is used for this purpose. The region size is an important parameter that determines the cost of group key management. Our proposed region-based group key management scheme will operate at the optimal region size identified to minimize the cost of key management in terms of network traffic.

Bootstrapping: In the initial bootstrapping process, a node within a region can take the role of a regional "leader" to perform *GDH*. If there are multiple initiators, then the node with the smallest *id* will win as the leader and will execute GDH to completion to generate a regional key. A key agreement protocol such as GDH is robust such that if a node leaves or moves out of the region during the execution of GDH, the remaining nodes can re-execute the protocol to completion and eventually agree on a shared secret key. Once a leader is generated in each region, all leaders in the group will execute GDH to agree on a secret *leader key*, K_{RL}, for secure communications among leaders. Similar with the bootstrapping process within a region, a "leader" can take the role of a "super-leader" or "coordinator" to execute GDH among leaders. If there are multiple leaders initiating the execution of GDH, the leader with the smallest *id* will win as the coordinator to execute GDH to completion to generate K_{RL}. Once K_{RL} is generated, a *group key*, K_G, is derived by means of $K_G = \text{MAC}(K_{RL}, c)$, where MAC is a cryptographically secure hash function, K_{RL} is the leader key used as the secret key to MAC, and c is a fresh counter which will be incremented whenever a group membership event occurs. Once K_G is generated, leaders will disseminate the group key K_G to group members in their regions. The group key K_G then is used for secure data communications among group members across regions. We note that a leader is responsible for first generating a new group key based on the current leader key (agreed upon by all leaders)

Table 1
Notation

Symbol	Meaning
K_G	Group key
K_{RL}	Leader key
K_{Ri}	Regional key in region i
RV_i	Regional view in region i
LV	Leader view
GV	Group view
RL_i	A leader in region i
$RM_{i,j}$	A member j in region i

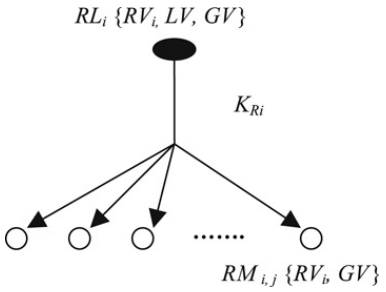


Fig. 2. Views for leaders and members.

and then broadcasting the renewed group key to members in its region. All leaders do the same, so the renewed group key may be disseminated to all members in all regions. In this sense, the leader is carrying out a centralized rekeying protocol in its region since the generation of the renewed key is done by the leader alone based on the current leader key without having to coordinate with other leaders.

Key management: These shared secret keys at the subgroup (regional), leader, group levels may be rekeyed to preserve secrecy in response to events that occur in the system. The *leader key* (K_{RL}) is rekeyed whenever there is a leader change, including a leader crossing a regional boundary or leaving the group, a leader failure, and a group merge or partition event. The *regional key* (K_R) is rekeyed whenever there is a regional membership change, including a local member group join/leave, a node failure, a local regional boundary crossing, and a group merge or partition event.

View management: In addition to maintaining secrecy, our region-based key management scheme also allows membership consistency [9] to be maintained through *membership views*. Three membership views can be maintained by various parties: (a) *Regional View (RV)* contains regional membership information including regional (or subgroup) members' *id*s and their location information, (b) *Leader View (LV)* contains leaders' *id*s and their location information, and (c) *Group View (GV)* contains group membership information that includes members' *id*s and their location information.

Fig. 2 illustrates the views maintained by a leader vs. those by a group member. A view can be established after a shared secret key is established at the corresponding regional, leader or group level.

Rekeying protocol: In addition to group member join/leave events which cause rekeying of the group key, mobility-induced events may also cause rekeying. Below we describe our region-based key management protocol for a MANET in response to events that may occur in the system.

– *Group member join*: All nodes are equipped with GPS, so a node knows which region it is in and which regional boundary it crosses. A join operation is initiated by a node contacting its neighbour nodes to know the leader node of the region in a group. When a new member, say A, joins the group, A beacons a "*hello*" message including its *id* and location information to inform its intention to join the group. Neighbouring nodes receiving the beacon forward the "*hello*" message to their regional leader. The regional leader authenticates A's identity based on A's public key. Then, the leader acts as a coordinator involving all subgroup members including A to execute GDH to generate a new regional key. The leader then updates the regional membership list, and broadcasts the regional membership list to members in the region. Since a join event incurs a group membership change, the group key

is also rekeyed. The regional leader informs the newly joined member's information to all other leaders so that all leaders apply $K_G = \text{MAC}(K_{RL}, c)$ to generate a new group key, using the current leader key K_{RL} as the secret key to MAC. All leaders then simultaneously distribute the new group key to members in their regions by encrypting the group key with their respective regional key K_R. In summary, when a new member joins the group, a regional key and a group key are rekeyed. The regional view of the region in which the group join event is initiated and the group view are updated.

- *Group member leave*: When a nonleader member, say B, leaves the group, it informs its leaving intention to its regional leader. When the leader receives the leaving intention message from B, it updates its regional view and disseminates the updated regional view to its members. Since a group leave event instigates a regional membership change event, a new regional key is generated by executing GDH and distributed to the regional members. Next, the leader informs the membership change information to all other leaders. After all leaders receive the information on the current leave event, they also broadcast the changed group view to all their members. Finally, all leaders autonomously regenerate a group key and distribute it to their corresponding members by encrypting the group key with their respective regional key K_R.
- *Group member leave by a leader member*: When a leader (who is also a member) leaves the group, a leader key also should be changed. Thus, in addition to all operations required in the above case for the nonleader member leave, a new leader is elected to replace the leaving leader. Since this involves a leader membership change, all leaders including the newly elected leader will execute GDH to generate a new leader key. Then each leader autonomously generates a new group key and distributes it to its members using the respective regional key K_R.
- *Boundary crossing by a non-leader member*: If a non-leader member crosses a regional boundary, for example, from region i to region j, a regional membership change occurs in both regions i and j. Thus, the regional keys in the two involved regions are respectively rekeyed based on GDH and the members' regional views in these two regions are updated. Since the mobility event changes neither the leader view nor the group view, no leader or group view updates are necessary. No rekeying of the group key is needed because the member leaving a region (subgroup) is still a member of the group.
- *Boundary crossing by a leader member*: If a leader member crosses a regional boundary from i to j, there is a leadership change in addition to all operations considered in the event of boundary crossing by a nonleader member. Thus, as in the group member leave by a leader member event, a new leader in the departing region is elected, the leader key is rekeyed among all leaders, and the leader view is updated among all leaders.
- *Group member disconnection and reconnection:* Members may be disconnected voluntarily (i.e. turn power off for energy saving) or involuntarily (i.e. obstructions or jamming, etc.). To detect a member failure in the group, each mobile host periodically sends an "I-am-alive" beacon message to its leader so that the leader is aware of which members are in its region. If a leader does not hear the beacon for a certain period of time (T) from a member, it considers the member being disconnected. Such disconnections are treated as group leave events in our protocol. If the member being disconnected is a leader, a new leader is elected by following a new leader election protocol. Temporarily disconnected member nodes can be later reconnected and rejoin the group. Reconnections are treated as group join events.
- *Leader election*: A group leave, a boundary crossing, or a disconnection by a leader member triggers a new leader election in the involved region. A member in the region after missing its regional leader's beaconing message can initiate the execution of GDH based on its regional view. If there are more than one leader invoking GDH, the member with the smallest *id* wins and will execute GDH to completion to generate a new regional key K_R. The new leader then announces itself as a new leader in the region by broadcasting a beacon message "I-am-a-new-leader" along with the new regional view encrypted with the regional key K_R.
- *Group partition*: Group members may not be able to communicate with each other because of group partitioning due to node failure and node mobility. Thus, a group may be partitioned into multiple groups dynamically. Group partitioning rate increases as node density decreases and as node mobility increases. A group partitioning event starts with a region being partitioned and members in a region miss beacon messages of each other. It is detected by members in a region missing the leader's beacon message, and by the leader's missing its regional members' beaconing messages. In the former case, a new leader is elected following the "leader election" protocol discussed earlier. In the latter case, the leader with the remaining nodes in the partitioned region will execute GDH to agree on a new regional key K_R as if the nodes in the other partitioned region had been disconnected. In either case, in each partitioned group, all leaders will execute GDH to agree on a new leader key K_{RL} following the *"group*

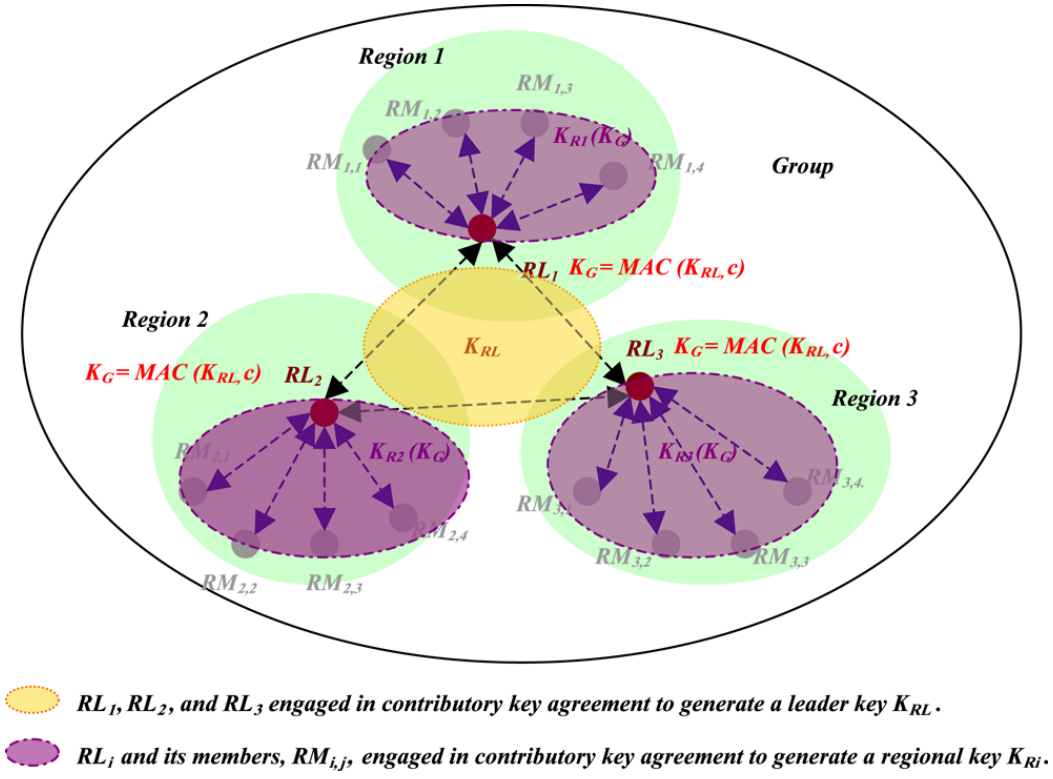


Fig. 3. Key generation and distribution.

member leave by a leader member" protocol discussed earlier, as if all leaders in the partitioned group had left the group.

– *Group merge*: Two groups may merge into one when connectivity resumes. A group merge event is detected by members within a region detecting the presence of beacon messages by nongroup members. After authentication (through the two leaders), members in the merged region will execute GDH to agree on a new regional key K_R following the "*group member join*" protocol as if members in the merged region had just newly joined the group. The new leader in the merged region then coordinates with all other leaders to execute GDH to agree on a new leader key K_{RL} as if leaders had been reconnected. Finally, a new group key is generated by all leaders and is distributed to all group members in the merged group.

Fig. 3 illustrates how our protocol generates and distributes regional, leader and group keys at the regional, leader and group-levels, respectively. The scenario shows that there are three regions in a group, each with a regional key $K_{Ri,}$ generated through the execution of a contributory key agreement by the leader RL_i with its members $RM_{i,j}$ in the region. Subsequently, RL_1, RL_2, and RL_3 are engaged in contributory key agreement to generate a leader key K_{RL}. The group key is then generated by MAC(K_{RL}, c).

4. Background

4.1. System model

Assume that nodes are randomly distributed according to a homogeneous spatial *Poisson* process with density λ_p. Assume that the operational area is $A = \pi r^2$, where r is the radius of the operational area. Thus, the average number of nodes in the system is $N = \lambda_p A$. Assume that a node may leave a group voluntarily with rate μ and may rejoin any group with rate λ due to tactical reasons. Then, the probability that a node is in any group is $\lambda/(\lambda + \mu)$ and the probability that it is not in any group is $\mu/(\lambda+\mu)$. It follows that the average number of nodes that are active members is given by $N\lambda/(\lambda + \mu)$. Furthermore, let Λ_J and Λ_L be the *aggregate* join and leave rates of all nodes, respectively. Then, Λ_J and Λ_L, can be calculated as:

$$\Lambda_J = \lambda \times N \times \frac{\mu}{(\lambda + \mu)} \qquad \Lambda_L = \mu \times N \times \frac{\lambda}{(\lambda + \mu)}. \tag{1}$$

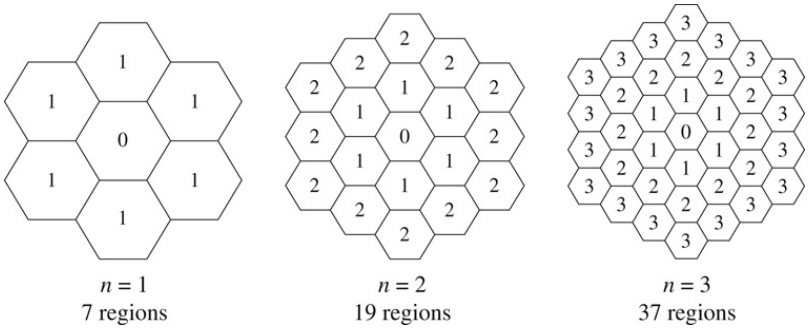


Fig. 4. A geographical area divided into $3n^2 + 3n + 1$ hexagons with $n = 1, 2$, and 3.

Nodes can move freely with a mobility rate of σ. Nodes that are connected with each other form a group. When all nodes are connected, there is only a single group in the system. Due to node mobility, a group may be partitioned into two. Conversely, two groups may merge into one as connectivity resumes. We assume that the secure group communication system is designed to support a mission critical application. All nodes are charged to complete a *mission* and the mission critical application allows group merging and partitioning activities in response to network dynamics. However, a group, no matter of its size, acts independently of other groups to complete the mission. Nodes in a group must satisfy the *forward/backward secrecy*, *confidentiality, integrity* and *authentication* requirements for secure group communications in the presence of malicious outsider attacks. Reliable transmission is a system requirement for secure group communications. We assume that *view synchrony (VS)* is guaranteed in group communication systems [1,3–5,42]. *VS* guarantees that messages are delivered reliably and in proper order under the same membership view. That is, a receiver will see the same membership view as viewed by the sender.

We use a hexagon to model a region. Fig. 4 shows a case in which the operational geographical area $A = \pi r^2$ is divided into $3n^2 + 3n + 1 = 37$ regions with $n = 3$, 19 regions with $n = 2$, and seven regions with $n = 1$, where n is the ring level. Each region has the same size. We do not consider different region sizes or adaptive region sizes. Let $R(n)$ denote the number of regions in the operational area. The expression for $R(n) = 3n^2 + 3n + 1$ is derived by mathematical induction based on the hexagonal network coverage model used.

A member can move around by crossing boundaries between regions. Assume that members are always confined in the geographical area of πr^2 divided into $R(n) = 3n^2 + 3n + 1$ regions as in a battlefield situation. The total number of regional boundary edges is $6(3n^2 + 3n + 1)$ counting internal edges twice to include reverse traffic. The total number of outward boundary edges surrounding the geographical area is $12n + 6$. Therefore, the probability that a member moves across a boundary between two regions (but not going out of the geographical area) once a move is made, $P_{RM}(n)$, is given by:

$$P_{RM}(n) = \frac{6(3n^2 + 3n + 1) - (12n + 6)}{6(3n^2 + 3n + 1)}. \tag{2}$$

The mobility model of a node remains the same. However, the mobility rate of a node, defined as the rate at which a region is crossed, changes depending on the number of regions $R(n)$ in the operational area. Let the mobility rate of a node be σ when there is only one region. As we divide the area into more regions (i.e. from $R(n) = 1, 7$ to 19, and so on as we increase n from 0, 1 to 2), the "regional" mobility rate increases because as the region size decreases causing more boundary-crossing events to occur per time unit. Let σ_n be the *regional* mobility rate when there are $R(n)$ regions. Then $\sigma_n = (2n+1)\sigma P_{RM}(n)$ because a node would cross $2n+1$ regions when there are $R(n)$regions for the same amount of time it would take to cross a regional boundary when there is only one region in the geographical area. The factor $P_{RM}(n)$ is multiplied to account for the fact that not all moves will cross a regional boundary. The relationship between n, $R(n)$ and σ_n is summarized below:

$$\begin{array}{lll} \text{level } 0 & 1 \text{ hexagon} & \sigma_0 = \sigma \\ \text{level } 1 & 7 \text{ hexagons} & \sigma_1 = 3\sigma P_{RM}(1) \\ \text{level } 2 & 19 \text{ hexagons} & \sigma_2 = 5\sigma P_{RM}(2) \\ \vdots & \vdots & \vdots \\ \text{level } n & (3n^2 + 3n + 1) \text{ hexagons} & \sigma_n = (2n + 1)\sigma P_{RM}(n). \end{array} \tag{3}$$

4.2. Security model

Our region-based group key management protocol must satisfy the *secrecy*, *confidentiality, integrity* and *authentication* requirements for secure group communications in the presence of malicious outsider attacks. Recall that (in Section 3) that a group key is generated by all leaders by applying a cryptographically secure MAC function using the leader key as a MAC key. Thus, *group key secrecy* is guaranteed since it is computationally infeasible for an adversary to discover the group key without knowing the secret key to MAC, which in our scheme is the leader key. *Forward* and *backward secrecy* [4] properties are preserved by means of immediate rekeying, i.e. a rekeying operation is performed whenever there is a membership change. *Forward secrecy* is guaranteed since a passive adversary who knows a contiguous subset of old group keys cannot discover any subsequent group key. *Backward secrecy* is guaranteed since a passive adversary who knows a contiguous subset of group keys cannot discover previous group key. Finally, *key independence* is guaranteed since a group key is generated using MAC with two different inputs, a leader key and a fresh counter, which guarantee key independence.

Each member has a private key and its certified public key available for *authentication* purposes. When a new member joins a group, the new member's identity is authenticated based on the member public/private key pair by applying the challenge/response mechanism. Source authentication is ensured during regional, leader and group key generation. When a regional key or a leader key is generated through GDH, the source authentication of a participating member is achieved by using the private/public key pair. *Confidentiality* is ensured by using the secret key shared by involved parties to encrypt information exchanged among the parties. Specifically, a regional key is only used by members within a region; a leader key is used among leaders; and a group key is only used by group members for group communication activities. When a group key is generated from the leader key, i.e. $K_G = \mathrm{MAC}(K_{RL}, c)$, based on MAC using the current leader key K_{RL} and a fresh counter c, the group key is encrypted by the regional key for a leader to disseminate to its members in the region. Thus, confidentiality is maintained because only members in the region have and can use the shared regional key to decrypt the group key.

Finally, to preserve *integrity*, a member can generate a MAC code using the secret key it shares with other group members as the MAC key when a message is sent. It is computationally impossible to alter the content of the message without being detected by a receiver that shares the secret key. This can be applied for intra-regional communication between a leader and its regional members using the shared regional key, for inter-regional communication among leaders using the shared leader key, or among members using the shared group key.

4.3. Attack model

MANETs pose many unique challenges for security design, including open network architecture, shared wireless medium, stringent resource restrictions, and highly changing network topology, etc. [30]. These challenges provide special opportunities for adversaries to attack MANETs. This region-based group key management scheme deals with *outsider attacks*, and specifically it can deal with *loss of authentication*, *loss of integrity*, and *loss of confidentiality* due to outsider attacks initiated from unauthorized or illegitimate users from the system [31]. In general, an outside attacker would attempt access to an authorized account and then perpetrate as an inside attacker. Below we discuss possible outsider attacks and countermeasures taken in our region-based group key management scheme:

- An outside attacker can gain unauthorized access to a legitimate account by eavesdropping data packets or any message containing a secret key for more sophisticated attacks. This attack is prevented by maintaining confidentiality using a secret key only known to authenticated participants. In particular, a new group key is encrypted with the secret regional key to disseminate to group members in each region, so it is impossible for an outside attacker not knowing the regional key to know the group key.
- An outside attacker can attempt to modify a data packet to break data integrity. This is prevented by generating a MAC code using the secret key as the MAC key when a message is sent. It is computationally impossible to alter the content of the message without being detected by a receiver that shares the secret key.
- An attacker may impersonate a group member to join a group. However, the attacker would not pass source authentication because it will fail the challenge/response test.
- An attacker may forge packets. However, since the attacker does not know the secret key, a forged packet will be discarded by the receiver because the packet cannot be properly decrypted by the receiver with the secret key. A replay attack can be prevented by incorporating a sequence number into each packet.

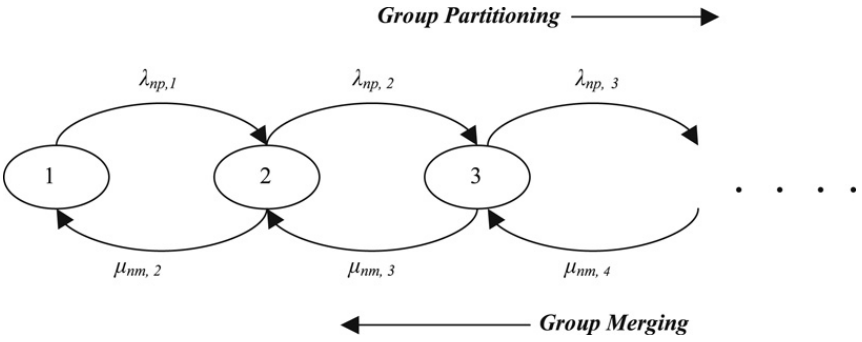


Fig. 5. A birth–death process for modelling group merging/partitioning.

5. Performance model and analysis

We develop a performance model to evaluate the network traffic cost generated for group key management in the proposed region-based protocol for MANETs. The goal of the performance analysis is to identify the optimal region size that will minimize the network traffic generated. The basic idea is to utilize the performance model developed to derive a formula to calculate the generated network traffic as a function of the region size, from which we could decide the best region size to minimize the network traffic, when given a set of basic parameter values characterizing the network and operational conditions.

The cost metric used for measuring the proposed group key management protocol is the *total network traffic per time unit* incurred in response to group key management events including regional mobility induced, group join/leave, periodic beaconing, and group merge/partition events. The "*cost*" refers to the amount of *information bits* multiplied by the number of hops these information bits travel, i.e. hop-bits. Thus, the total cost ($\hat{C}_{\text{total}}$) consists of four components:

- *Group merge/partition cost* ($\hat{C}_{mp}$): This is the cost per time unit for dealing with group partitioning and merging events. Whenever a group partitioning/merging event occurs, it is required to rekey the group key and update the view for involved partitioned/merged groups such that the secrecy requirements are satisfied.
- *Regional mobility cost* ($\hat{C}_{\text{mobility}}$): This is the cost per time unit in response to mobility-induced regional boundary crossing events, or *regional mobility handoff* events.
- *Group join/leave cost* ($\hat{C}_{\text{join/leave}}$): This is the cost per time unit for handling group join or leave events. This cost also includes the cost caused by connection/disconnection events by group members.
- *Periodic beaconing cost* ($\hat{C}_{\text{beacon}}$): This is the cost per time unit for maintaining view consistency by all members through periodic beaconing. Thus, this cost includes the cost for broadcasting periodic beaconing messages such as "I-am-alive," "I-am-a-new-leader," etc. By using this mechanism, member connection or disconnection events can be detected.

The magnitude of these cost components actually depends on how many groups exist when an event occurs. To this end, we first decide the steady state probability of the system having i groups. Then, we calculate the *average* cost based on the steady-state probability. We use a *birth–death* process shown in Fig. 5, to model the system. We assume that group merging and partitioning events follow the one-event assumption, that is, they occur one at a time.

In Fig. 5, each state i represents i partitioned groups, at which the merging and partitioning rates are state dependent and are represented by $\mu_{nm,i}$ and $\lambda_{np,i}$, respectively. These state-dependent merging/partitioning rates essentially depend on node density, number of groups (that is, i), and node mobility. We will explain how we parameterize these state-dependent merging/partitioning rates in Section 6. Once we parameterize the birth–death model, the probability of the system being in state i, $\Pr_i$, can be easily calculated from queuing theory.

The total cost ($\hat{C}_{\text{total}}$) incurred by the region-based key management protocol is calculated as follows:

$$\hat{C}_{\text{total}} = \hat{C}_{\text{join/leave}} + \hat{C}_{\text{mobility}} + \hat{C}_{\text{beacon}} + \hat{C}_{mp} \tag{4}$$

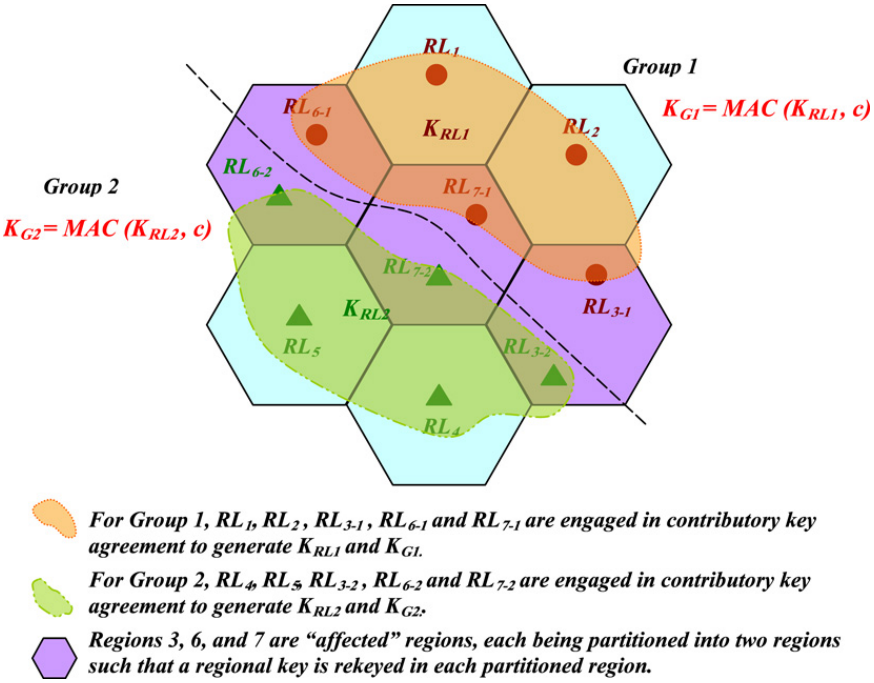


Fig. 6. Dynamic reconfiguration in response to group partitioning under $R(1) = 7$.

where:

$$\begin{aligned}
\hat{C}_{\text{join/leave}} &= \sum_{i=1}^{\infty} \Pr_i \times \hat{C}_{\text{join/leave},i} \\
\hat{C}_{\text{mobility}} &= \sum_{i=1}^{\infty} \Pr_i \times \hat{C}_{\text{mobility},i} \\
\hat{C}_{\text{beacon}} &= \sum_{i=1}^{\infty} \Pr_i \times \hat{C}_{\text{beacon},i} \\
\hat{C}_{mp} &= \sum_{i=1}^{\infty} \Pr_i \times \hat{C}_{mp,i}
\end{aligned} \tag{5}$$

where $\Pr_i$ is the steady-state probability that the system is in state i (i.e. the number of groups is i); $\hat{C}_{\text{join/leave},i}$, $\hat{C}_{\text{mobility},i}$, $\hat{C}_{\text{beacon},i}$, and $\hat{C}_{mp,i}$, are the respective cost components, given that the number of groups in the system is i. Note that in the special case that the density is sufficiently high such that nodes are connected all the time, the group partitioning rate is zero, so the merge/partition cost is also zero. Below we explain how we calculate $\hat{C}_{\text{join/leave},i}$, $\hat{C}_{\text{mobility},i}$, $\hat{C}_{\text{beacon},i}$ and $\hat{C}_{mp,i}$. Basic parameters in the model are summarized in Table 2. Table 3 gives parameters derived from basic parameters.

5.1. Cost for group merging and partitioning: $\hat{C}_{mp,i}$

The traffic cost incurred per time unit due to group partitioning/merging while the system in state i, $C_{mp,i}$, is the sum of that due to partitioning, $C_{\text{partition},i}$, and that due to merging, $C_{\text{merge},i}$. We observe that $C_{\text{merge},1,} = 0$ since in state 1 there is no merging event.

Fig. 6 illustrates a group partitioning event for the case in which the number of hexagon regions is 7. Initially the system contains one group. Later the group is split into two groups, 1 and 2. Regions 3, 6, and 7 are the ones on which group partitioning occurs. As a result, regions 3, 6 and 7 are each partitioned into two regions, e.g. region three is partitioned into region 3-1 and region 3-2, and region six is partitioned into regions 6-1 and 6-2, and so on. In each of these "partitioned" regions, a regional key needs to be rekeyed since the membership in the region has been changed.

Table 2
Basic model parameters

Parameter	Meaning
Σ	Mobility rate per node
λ	Group join rate per node
μ	Group leave rate per node
$\lambda_{np,i}$	Per-group group partition rate when the number of groups $= i$
$\mu_{nm,i}$	Per-group group merge rate when the number of groups $= i$
$\lambda_p A$	Node density (nodes/km^2) operational area of the mobile group, that is, $A = \pi r^2$(km^2) where r is the radius
A_{region}	Area of a region
s	Radius of a hexagon region
r	Radius of the operational area
R	Wireless per-hop radio range (m)
k	Size of a group key (bits)
v	Size of each intermediate value in GDH (bits)
T_{RB}	Intra-regional beaconing interval (s)
T_{LB}	Inter-regional beaconing interval (s)
M_{alive}	Size of a beacon message (bits)
M_{pub}	Size of a publishing message (bits)
U_{view}	Size of a view update message (bits)
$C_{np,i}$	Cost per group partitioning event when the number of groups $= i$ (hop bits/s)
$C_{nm,i}$	Cost per group merging event when the number of groups $= i$ (hop bits/s)
C_{intra}	Cost for intrakey rekeying and regional view updating in a region (hop bits/s)
$C_{\text{inter},i}$	Cost for inter-key rekeying and leader view updating in a group (hop bits/s)
H_{region}	Number of hops between a leader and a member within a region
$H_{\text{leader},i}$	Number of hops among leaders in a group when the number of groups $= i$
$R(n)$	Number of regions in the system
$N_{\text{region},i}$	Number of regions in a group when the number of groups $= i$
N_{np}^{region}	Number of partitioned regions in a group after a group partitioning event
N_{nm}^{region}	Number of merged regions in a group after a group merging event
$N_{\text{region}}^{\text{members}}$	Number of members in a region
$C_{\text{rekey}}^{\text{intra}}$	Intra-regional cost for rekeying a regional key (hop bits/s)
$C_{\text{update}}^{\text{intra}}$	Intra-regional cost for updating a regional view (hop bits/s)
$C_{\text{rekey},i}^{\text{inter}}$	Inter-regional cost for rekeying a leader key when the number of groups $= i$ (hop bits/s)
$C_{\text{update},i}^{\text{inter}}$	Inter-regional cost for updating a leader view when the number of groups $= i$ (hop bits/s)
$C_{\text{leader},i}^{\text{change}}$	Cost for a leader change in a group when the number of groups $= i$ (hop bits/s)

Table 3
Derived parameters

Parameter	Meaning
σ_n	Regional mobility rate per node
Λ_J	Aggregate group join rate
Λ_L	Aggregate group leave rate
Λ_{RB}	Aggregate periodic beaconing rate from all members in the system
Λ_{LB}	Aggregate periodic beaconing rate from all leaders in the system

On the other hand, regions 1 and 2 in group 1 as well as regions 4 and 5 in group 2 are not affected by the group partitioning event, so the regional key needs not be rekeyed in these regions. After partitioning, group 1 contains 5 regions, labelled as RL_1, RL_2, $RL_{3\text{-}1}$, $RL_{6\text{-}1}$ and $RL_{7\text{-}1}$, while group 2 also contains 5 regions, labelled as RL_4, RL_5, $RL_{3\text{-}2}$, $RL_{6\text{-}2}$ and $RL_{7\text{-}2}$. In group 1, RL_1, RL_2, $RL_{3\text{-}1}$, $RL_{6\text{-}1}$ and $RL_{7\text{-}1}$ then execute a contributory key agreement protocol to generate K_{RL1} and subsequently K_{G1}, while in group 2, RL_4, RL_5, $RL_{3\text{-}2}$, $RL_{6\text{-}2}$ and $RL_{7\text{-}2}$ are engaged in contributory key agreement to generate K_{RL2} and subsequently K_{G2}.

After understanding the cost involved due to a group partitioning event, we calculate $C_{\text{partition},i}$ as the product of the group partitioning rate at state i, $\lambda_{np,i}$, and the cost per group partitioning event in state i, $C_{np,i}$, i.e.

$$C_{\text{partition},i} = \lambda_{np,i} \times C_{np,i} \tag{6}$$

$$C_{np,i} = 2 \times \left[N_{np}^{\text{region}} \times C_{\text{intra}} + C_{\text{inter},i} + C_{\text{group},i} + C_{\text{leader},i}^{\text{change}} \right]. \tag{7}$$

Here $C_{np,i}$ covers four costs: an intra-regional cost for rekeying a regional key in each of the "partitioned" regions in a group ($N_{np}^{\text{region}} \times C_{\text{intra}}$), an inter-regional cost for rekeying a leader key in each partitioned group ($C_{\text{inter},i}$), a cost for rekeying a group key in each partitioned group ($C_{\text{group},i}$), and a cost for changing leader in each partitioned group ($C_{\text{leader},i}^{\text{change}}$). Since a group partitioning event results in two partitioned groups, the incurred cost is multiplied by two. Note that C_{intra}, $C_{\text{inter},i}$, $C_{\text{group},i}$, and $C_{\text{leader},i}^{\text{change}}$ are each a per-group cost at state i, as given in Eqs. (14), (17), (23) and (19) respectively. Also N_{np}^{region} is the number of "partitioned" regions in a group after group partitioning; it is equal to one if the group size is smaller than the region size; otherwise it is equal to the ratio of the group size to the region size, viz.

$$\begin{aligned} &\text{if } (r/\sqrt{i} > s) \\ &N_{np}^{\text{region}} = \frac{r}{s\sqrt{i}}; \\ &\text{else} \\ &N_{np}^{\text{region}} = 1; \end{aligned} \tag{8}$$

where $r/\sqrt{i}$ is the radius of a group after group partitioning and s is the radius of a hexagonal region.

Next we compute the traffic cost incurred per time unit due to group merging while the system in state i, $C_{\text{merge},i}$; it is computed by the product of the group merging rate at state i,$\mu_{nm,i}$, and the cost per group merging event in state i, $C_{nm,i}$, i.e.

$$C_{\text{merge},i} = \mu_{nm,i} \times C_{nm,i} \tag{9}$$

$$C_{nm,i} = \left(N_{nm}^{\text{region}} \times C_{\text{intra}} \right) + C_{\text{inter},i} + C_{\text{group},i} + C_{\text{leader},i}^{\text{change}}. \tag{10}$$

Here the cost per group merge event in state i ($C_{nm,i}$) covers 4 cost components: an intra-regional cost in the merged group ($N_{nm}^{\text{region}} \times C_{\text{intra}}$), an inter-regional cost in the merged group ($C_{\text{inter},i}$), a cost for rekeying a group key and updating a group view in the merged group ($C_{\text{group},i}$), and a cost for changing leader in the merged group ($C_{\text{leader},i}^{\text{change}}$). Similar to the calculation of N_{np}^{region}, N_{nm}^{region} is calculated based on Eq. (8) except that the radius of the group after group merging is being used in the calculation.

Summarizing the above, the cost per time unit due to group partitioning/merging events while the system in state i, $\hat{C}_{mp,i}$, is the sum of that due to partitioning and that due to merging:

$$\hat{C}_{mp,i} = C_{\text{partition},i} + C_{\text{merge},i}. \tag{11}$$

5.2. Cost for regional boundary crossing: $\hat{C}_{\text{mobility},i}$

The traffic cost incurred per time unit due to a regional boundary event while the system in state i, $\hat{C}_{\text{mobility},i}$, covers two cases: (a) a boundary crossing by a nonleader; (b) a boundary crossing by a leader. Thus, $\hat{C}_{\text{mobility},i}$ is given by:

$$\hat{C}_{\text{mobility},i} = \Lambda_m \times \left[C_{\text{mobility}}^{\text{nonleader}} + C_{\text{mobility}}^{\text{leader}} \right]. \tag{12}$$

Here Λ_m is the aggregate regional mobility by nodes in the system, given by $\sigma_n \times N$. The cost for the system to handle a nonleader member crossing a regional boundary is given by

$$C_{\text{mobility}}^{\text{nonleader}} = P_{\text{nonleader}} \times [C_{\text{intra}} \times 2]. \tag{13}$$

Here $P_{\text{nonleader}}$ is the probability of a nonleader given by $P_{\text{nonleader}} = (N - N_{\text{leader}})/N$ where N is the total number of nodes and N_{leader} is the number of leaders in the system, C_{intra} is the cost incurred for rekeying K_R and updating the regional view in a region, given in Eq. (14) below:

$$C_{\text{intra}} = \left[C^{\text{intra}}_{\text{update}} + C^{\text{intra}}_{\text{rekey}}\right] \times H_{\text{region}} \tag{14}$$

where $C^{\text{intra}}_{\text{update}}$ is the cost for updating a regional view, $C^{\text{intra}}_{\text{rekey}}$ is the cost for rekeying a regional key, and H_{region} is the number of hops within a region for a regional leader to disseminate a regional view or key to the members in its region, given by Eq. (15):

$$H_{\text{region}} = \frac{s}{R} \quad s = \sqrt{\frac{2}{3\sqrt{3}} A_{\text{region}}} \tag{15}$$

$$A_{\text{region}} = \frac{A}{R(n)}.$$

Here A_{region} is the area of a region, s is the circum-radius of a hexagon-shaped region, and R is the wireless per-hop radio range. When there is no region, that is, when $R(n) = 1$ at $n = 1$, $A_{\text{region}} = A$.

On the other hand, the cost for handling a leader member boundary crossing event is:

$$C^{\text{leader}}_{\text{mobility},i} = P_{\text{leader}} \times \left[C_{\text{intra}} \times 2 + C_{\text{inter},i} + C^{\text{change}}_{\text{leader},i}\right] \tag{16}$$

where $C_{\text{inter},i}$ is the cost for rekeying a leader key and updating the leader view, given below in Eq. (17), $P_{\text{leader}} = R(n)/N$ where $P_{\text{leader}} = 1 - P_{\text{nonleader}}$ is the probability of a leader crossing a regional boundary, and $C^{\text{change}}_{\text{leader},i}$ is the cost for changing a leader in a region, given below in Eq. (19).

The cost for inter regional communications ($C_{\text{inter},i}$) is computed as

$$C_{\text{inter},i} = \left[C^{\text{inter}}_{\text{update},i} + C^{\text{inter}}_{\text{rekey},i}\right] \times H_{\text{leader},i} \tag{17}$$

where $C^{\text{inter}}_{\text{update},i}$ is the cost for updating the leader view in a group, $C^{\text{inter}}_{\text{rekey},i}$ is the cost for rekeying the leader key in a group, and $H_{\text{leader},i}$ is the number of hops among leaders for a leader to disseminate a leader view or key to other leaders in a group. Since there are i groups in the system, the radius of a group can be approximated as $r/\sqrt{i}$ where r is the radius of the operational area. Consequently, the number of hops among leaders in a group, $H_{\text{leader},i}$, is given by:

$$H_{\text{leader},i} = \frac{r}{R\sqrt{i}}. \tag{18}$$

For $C^{\text{change}}_{\text{leader},i}$, the outgoing leader would broadcast two messages announcing its intention to leave, with one message to its regional members using its regional key, and another message to other leaders using a leader key. In addition, the new leader would broadcast two messages expressing "I-am-a-new-leader" to its regional members and to the leader group using its regional key and the leader key, respectively. Further, these messages need to travel through a number of hops at the leader and intra-regional levels represented by $H_{\text{leader},i}$ and H_{region}, respectively. Thus, the cost for a leader change, $C^{\text{change}}_{\text{leader},i}$, is calculated as

$$C^{\text{change}}_{\text{leader},i} = H_{\text{leader},i} \times \left[M^{\text{leaders}}_{\text{old-leader}} + M^{\text{leaders}}_{\text{new-leader}}\right] + H_{\text{region}} \times \left[M^{\text{regional-members}}_{\text{old-leader}} + M^{\text{regional-members}}_{\text{new-leader}}\right]. \tag{19}$$

Summarizing above, $\hat{C}_{\text{mobility},i}$ is given by:

$$\hat{C}_{\text{mobility},i} = \Lambda_m \times \left\{[2 \times C_{\text{intra}}] + P_{\text{leader}} \times \left[C_{\text{inter},i} + C^{\text{change}}_{\text{leader},i}\right]\right\}. \tag{20}$$

5.3. Cost for group join/leave: $\hat{C}_{\text{join/leave},i}$

$\hat{C}_{\text{join/leave},i}$ includes the cost for handling group join and leave. Thus,

$$\hat{C}_{\text{join/leave},i} = \Lambda_J \times C_{\text{join},i} + \Lambda_L \times C_{\text{leave},i}. \tag{21}$$

Here Λ_J and Λ_L are the aggregate group join and leave rates of all members, respectively, given in Eq. (1). A group join event requires the update of the regional view and the rekeying of the regional key in the region from which the join event is originated, the cost of which is C_{intra}, as well as the update of the group view and the rekeying of a group key, the cost of which is $C_{\text{group},i}$. Therefore,

$$C_{\text{join},i} = C_{\text{intra}} + C_{\text{group},i} \tag{22}$$

where $C_{\text{group},i}$ is given by:

$$C_{\text{group},i} = C^{\text{group}}_{\text{update},i} + C^{\text{group}}_{\text{rekey},i} = \left[H_{\text{leader},i} \times M^{\text{leaders}}_{\text{update}} + H_{\text{region}} \times N_{\text{region},i} \times M^{\text{regional-members}}_{\text{update}}\right] + \left[H_{\text{region}} \times N_{\text{region},i} \times M^{\text{regional-members}}_{\text{rekey}}\right]. \tag{23}$$

Here $M^{\text{leaders}}_{\text{update}}$ is the number of bits required in a broadcast message for updating the group view for the leaders, $M^{\text{regional-members}}_{\text{update}}$ for updating the group view for members in a region, and $M^{\text{regional-members}}_{\text{update}}$ for rekeying the group key for members in a region; $N_{\text{region},i}$ is the number of regions in a group, given by $R(n)/i$.

The cost for group leave event includes two cases, namely, when a nonleader member leaves and when a leader leaves the group. Thus, the cost for a group leave event is:

$$C_{\text{leave},i} = C^{\text{nonleader}}_{\text{leave},i} + C^{\text{leader}}_{\text{leave},i} \tag{24}$$

with

$$C^{\text{non-leader}}_{\text{leave},i} = P_{\text{non-leader}} \times \left[C_{\text{intra}} + C_{\text{group},i}\right] \tag{25}$$

$$C^{\text{leader}}_{\text{leave},i} = P_{\text{leader}} \times \left[C_{\text{intra}} + C_{\text{inter},i} + C_{\text{group},i} + C^{\text{change}}_{\text{leader},i}\right] \tag{26}$$

where C_{intra}, $C_{\text{inter},i}$, $C_{\text{group},i}$, and $C^{\text{change}}_{\text{leader},i}$ are given earlier in Eqs. (14), (17), (23) and (19) respectively, and $P_{\text{leader}}$and $P_{\text{nonleader}}$ are as previously described. Here we note that the case in which a leader becomes disconnected, either voluntarily or involuntary, is considered as a leave event whose cost is given by Eq. (26), accounting for the cost for a leave event by a member (C_{intra} and $C_{\text{group},i}$) plus the cost for forming a new leader key ($C_{\text{inter},i}$) and the cost for a leader change ($C^{\text{change}}_{\text{leader},i}$).

5.4. Cost for periodic beaconing: $\hat{C}_{\text{beacon},i}$

$\hat{C}_{\text{beacon},,i}$ includes the cost of beaconing messages in two levels, namely, intra-regional beaconing among members in a region for maintaining the regional view, and inter-regional beaconing among leaders for maintaining the leader view. Thus, $\hat{C}_{\text{beacon},i}$ is computed as:

$$\hat{C}_{\text{beacon},i} = \left[\Lambda_{RB} \times M_{\text{alive}} \times H_{\text{region}}\right] + \left[\Lambda_{LB} \times M_{\text{alive}} \times H_{\text{leader},i}\right] \tag{27}$$

where M_{alive} is the number of bits in a beacon message, H_{region} is the number of hops between a regional leader and a regional member as given in Eq. (15), $H_{\text{leader},i}$ is the number of hops among leaders in a group when there are i groups as given in Eq. (18), and Λ_{RB} and Λ_{LB} are the overall beacon rates in the system by all of its members at the intra-regional level, and by all of its leaders at the inter-regional level, respectively. Λ_{RB} and Λ_{LB} are obtained from the reciprocals of the periodic beaconing intervals, T_{RB} and T_{LB}, at the intra-regional level and at the leader level, respectively, multiplied by the number of members in the operational area, $N\lambda/(\lambda + \mu)$, and the number of leaders

Stage 1: upflow	$M_1 \rightarrow M_2 \rightarrow \ldots \rightarrow M_{m-2} \rightarrow M_{m-1}$	
message size	$v \quad v \ldots\ldots v$	$= v(m-2)$
Stage 2: broadcast	$M_{m-1} \rightarrow M_i$ *where* $i \neq m\text{-}1$	
message size	v	$= v$
Stage 3: response	M_i *where* $i \neq m \rightarrow M_m$	
message size	v *from each* M_i	$= v(m-1)$
Stage 4: broadcast	$M_m \rightarrow M_i$ *where* $i \neq m$	
message size	$v(m-1)$ *intermediate values*	$= v(m-1)$
Total communication cost		$= 3v(m-1)$

Fig. 7. Message size requirement in each stage of GDH.3.

(which is the same as the number of regions), $R(n)$, respectively, i.e.

$$\Lambda_{RB} = N \times \left\lceil \frac{\lambda}{\lambda + \mu} \right\rceil \times \frac{1}{T_{RB}} \qquad \Lambda_{LB} = R(n) \times \frac{1}{T_{LB}}. \tag{28}$$

6. Example

6.1. Parameterization

Without loss of generality, we exemplify our region-based key agreement protocol with GDH [8] as the key agreement protocol to be used for secret key generation. We use it at both the intra-regional and inter-regional (leader) levels. Note that at the intra-regional level, since a leader within each region can take the role of a centralized coordinator, we may apply a centralized group key management protocol such as LKH [29] for intra-regional rekeying. However, we opt for fully distributed key management by using a distributed rekeying algorithm (e.g. GDH) at both levels. In particular, we adopt GDH.3 in [8] since GDH.3 allows the use of fixed-sized messages and only a constant (and smaller) number of exponentiation operations executed by each participant. With these features, GDH.3 has been proposed for mobile devices with low computational capabilities [22]. Below we briefly explain how GDH.3 works and how we parameterize $C^{\text{intra}}_{\text{rekey}}$ and $C^{\text{inter}}_{\text{rekey},i}$ as a function of the number of participants, m. Again, the cost is measured by the number of information bits required by GDH.3 multiplied by the number of hops information bits are transmitted.

GDH.3 comprises four stages. Each participant M_i shares a common base α and keeps its secret share N_i. The first stage collects contributions from all group members, $M_1, M_2 \ldots, M_m$. Specifically, M_1 raises α to the power of N_1, performing one exponential computation to generate α^{N1}, M_2 computes α^{N1N2} by raising α^{N1} to the power of N_2, and so on until M_{m-1} computes $\alpha^{N1 \ldots Nm-1}$. After processing the upflow message, M_{m-1} obtains $\alpha^{\prod\{N_k | k \in [1,m-1]\}}$ and broadcasts this value in the second stage to all other participants. In the third stage, every M_i factors out its own exponent and forwards the result to M_m. In the final stage, M_m collects all inputs from all other participants, raises every one of them to the power of $N_m$and broadcasts the resulting $m-1$ values to the rest of the group. Every M_i receives this message in the form of $\alpha^{\prod\{N_k | k \in [1,m-1] \wedge k \neq i\}}$ and can easily generate the intended secret key K_m.

Fig. 7 summarizes the number of bits required in each stage of GDH.3 where m is the number of participants and v is the size of each intermediate value. We apply GDH.3 as the key agreement protocol at both the intra-regional and inter-regional levels. In the former case, m is the number of members in a region; in the latter case, m is the number of leaders in a group.

In [8], the secrecy property of GDH has been proven. Since it is possible that join/leave operations may occur frequently in our target applications, we adopt an optimized way discussed in [22] with some modification to perform join/leave operations. More specifically, M_m saves the contents of the original broadcast and response messages from stages 2 and 3 in Fig. 7. When a new member, say, M_{m+1}, joins, it forwards its contribution to M_m. Then, M_m uses the new contribution along with contributions of existing members saved to generate a new set of subkeys, and distributes it to all members as described in stage 4 of Fig. 7. Therefore, a new member join process only requires two additional rounds. The *backward secrecy* is preserved because a new exponent is used in generating a new set of subkeys by M_m. For a member leave, if M_p is a member to be removed from the group (i.e. a member leave), M_m takes the special role of generating a new set of $n-2$ subkeys (excluding a portion of leaving member, M_p) by using a new exponent.

$$
\begin{aligned}
&Stage\ 1: unicast\\
&\qquad Number\ of\ hops \times message\ size = 1 \times v(N_{region}^{members} - 2)\\
&Stage\ 2: broadcast\\
&\qquad Number\ of\ hops \times message\ size = H_{region} \times v\\
&Stage\ 3: unicast\\
&\qquad Number\ of\ hops \times message\ size = H_{region} \times v\left(N_{region}^{members} - 1\right)\\
&Stage\ 4: broadcast\\
&\qquad Number\ of\ hops \times message\ size = H_{region} \times v\left(N_{region}^{members} - 1\right)
\end{aligned}
$$

Fig. 8. Parameterizing intra-regional communication cost ($C_{\text{rekey}}^{\text{intra}}$) based on GDH.

$$
\begin{aligned}
&Stage\ 1: unicast\\
&\qquad Number\ of\ hops \times message\ size = H_{leader}^{stage1(GDH)} \times v(N_{region,\,i} - 2)\\
&Stage\ 2: broadcast\\
&\qquad Number\ of\ hops \times message\ size = H_{leader,\,i} \times v\\
&Stage\ 3: unicast\\
&\qquad Number\ of\ hops \times message\ size = H_{leader,\,i}^{stage3(GDH)} \times v\\
&Stage\ 4: broadcast\\
&\qquad Number\ of\ hops \times message\ size = H_{leader,\,i} \times v\left(N_{region,\,i} - 1\right)
\end{aligned}
$$

Fig. 9. Parameterizing inter-regional communication cost ($C_{\text{rekey},i}^{\text{inter}}$) based on GDH.

Then, M_m distributes the new set of subkeys to all members. Finally, each member can generate a new group key using the new set of subkeys. Since the subkey for M_p is not included in the received subkeys, M_p cannot generate a new group key, thus preserving the *forward secrecy*. A member leave process only needs one round. If M_m is the member to be removed from the group, M_{m-1} takes the special role of distributing a set of subkeys. These optimized join and leave operations are mainly developed to reduce the number of rounds where bandwidth is a crucial issue. We use these optimized join/leave operations only if a leader is not involved in a join/leave operation.

Fig. 8 shows how $C_{\text{rekey}}^{\text{intra}}$ in the unit of hop-bits for rekeying a regional key is calculated after considering the number of hops information bits travel in each step of GDH. Here $N_{\text{region}}^{\text{members}}$ refers to the number of members in a region and H_{region} is calculated by Eq. (15). $C_{\text{rekey}}^{\text{intra}}$ is calculated as the total cost from these four stages. The cost of stage 1 is the number of bits required in stage 1, $v(N_{\text{region}}^{\text{members}} - 2)$ as shown in Fig. 8. Here the number of hops is 1 because in stage 1 of GDH, the message exchange is transmitted in a series fashion from M_1 to M_{n-1} and each node can communicate its neighbor with only one hop.[1] The cost of stage 2 is computed by the number of bits required for stage 2, namely, v, multiplied with the average number of hops in a region, H_{region}. The cost of stage 3 is the number of bits required, $v(N_{\text{region}}^{\text{members}} - 1)$, multiplied with the number of hops these bits travel, H_{region}. Here the number of hops is H_{region} because in stage 3 of GDH each member sends an intermediate value (v) to a regional leader. The cost of stage 4 is computed by the number of bits required for stage 4, namely, $v(N_{\text{region}}^{\text{members}} - 1)$, multiplied with the average number of hops in a region, H_{region}. Note that the cost for stage 3 is slightly overestimated by using the maximum number of hops that a message may travel.

Fig. 9 shows how $C_{\text{rekey},i}^{\text{inter}}$ for rekeying the leader key is computed. The computational steps are similar to those in Fig. 8. The number of participants (m) now is the number of leaders in a group which is the same as the number of regions in the group, $N_{\text{region},i} = R(n)/i$. The average number of hops between two leaders, $H_{\text{leader},i}$, is obtained based on Eq. (18) and is used in stages 2 and 4. The number of hops information bits travel between two leaders in stage 1 is given by:

$$H_{\text{leader}}^{\text{stage1(GDH)}} = 2s/R \tag{29}$$

[1] In stage 1, the chain can be dynamically formed by having each node pick a nearest node that has not been put on the chain to unicast the next message to. By this way two consecutive nodes on the chain would be mostly one-hop apart except for the last few nodes. Nevertheless, the network traffic cost for stage 1 is underestimated because the last few nodes on the chain could be multi-hop apart.

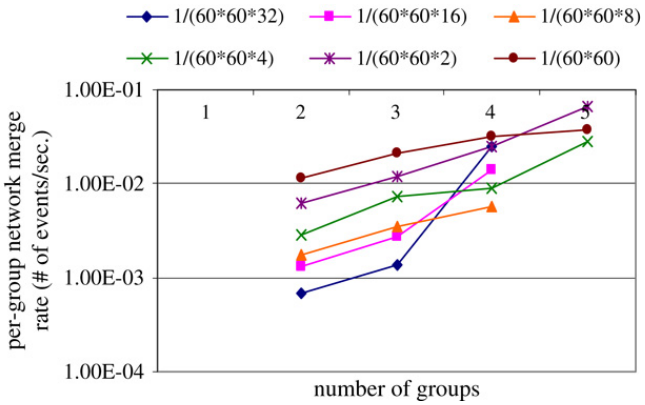


Fig. 10a. Group merging rate.

where s is the radius of a region calculated by Eq. (15) and R is the per-hop radio range in the wireless network. This computation is based on the assumption that the leader is located at the centre of each region. Thus, the distance between leaders in one ring level (n) of a hexagonal area and those in another is simply the diameter of a region. On the other hand, the number of hops information bits travel in stage 3 is given by

$$H_{\text{leader},i}^{\text{stage3(GDH)}} = \frac{\left(\sum_{k=1}^{n} 6k\,[2s/R]\right)}{i} \tag{30}$$

where n is the level of ring in the hexagonal area, s is the radius of a region, i is the number of groups observed, and R is the wireless radio range. The distance between a leader and another leader is approximately calculated as the diameter of a hexagon region. Eq. (30) means adding up all the number of hops by leaders in each level divided by the number of groups observed. In stage 3 of Fig. 9, since Eq. (30) already includes $N_{\text{region},i} - 1$ number of transmissions, the unicast cost is simply calculated by multiplying $H_{\text{leader},i}^{\text{stage3(GDH)}}$ by v, an intermediate message size.

Next we explain how we parameterize the per-group merging/partitioning rates at state i. While an analytical model is possible, we opt to run a simulation study to better reflect the dependency of the per-group merging/partitioning rates with node mobility and density. We collect the number of merging and partitioning events observed during a sufficiently long period of time T. We also sum the sojourn time in which i groups are observed in the system. Let S_i denotes this sojourn time that the system stays in state i (in which the number of groups $= i$). Let $N_{nm,i}$ and $N_{np,i}$ be the numbers of merging or partitioning events observed at the number of groups = i, respectively. Then, the merging/partitioning rates when the number of groups observed is i, represented by $\mu_{nm,i}$ and $\lambda_{np,i}$, are given by:

$$\mu_{nm,i} = \frac{N_{nm,i}}{S_i} \quad \lambda_{np,i} = \frac{N_{np,i}}{S_i}. \tag{31}$$

We use the measured $\mu_{nm,i}$ and $\lambda_{np,i}$ obtained above to parameterize the *birth–death* process model shown in Fig. 5. Fig. 10 shows the merging/partitioning rates collected from simulation as a function of regional mobility rate (σ) when the node density (λ_p) = 150, per-hop wireless radio range (R) = 200 m, and initial operational area (A) = πr^2 where r = 1 km with other default parameters listed in Table 4. Under these conditions, we observed only a maximum of five groups in simulation.

The general tendency observed here is that as the number of groups increases, the group partitioning rate decreases and the group merging rate increases. As shown in Fig. 10, the general trends show that as the regional mobility rate (σ) increases, higher per-group merging/partitioning rates are observed with the merging rate having a higher sensitivity over the partitioning rate. The trend also shows that as the regional mobility rate decreases, a group is less likely to be partitioned but groups are more likely to be merged, and vice versa.

6.2. Numerical analysis

Below we report numerical data for the communication cost incurred per time unit in executing the adopted region-based group key management protocol as a function of model parameters. We demonstrate that there exists an optimal

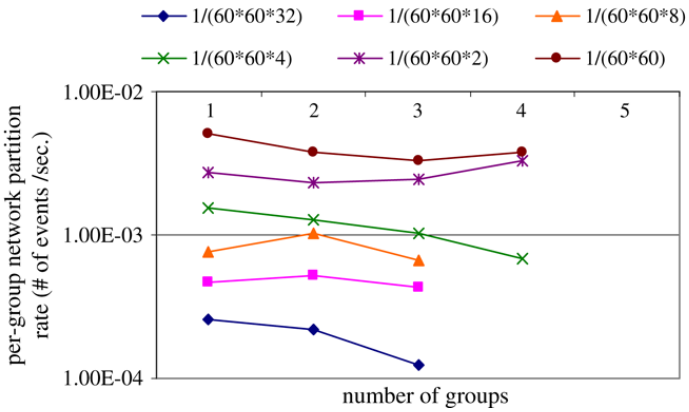


Fig. 10b. Group partitioning rate.

Table 4
Default parameter values

Parameter	Default value
σ	1/(60*60*32)
λ	1/(60*60) (once per hour)
μ	1/(60*60*4) (once per four hours)
λ_p	150 nodes/km^2
A	π km^2
R	200 m
k	64 bits
v	64 bits
T_{RB}	5 s
T_{LB}	2 s
U_{view}	500 bits
B_{alive}	32 bits

region size that minimizes the overall communication cost. The effect of region size is represented by a parameter, n, where $n = 0$ means that there is only one region (our baseline model), $n = 1$ means 7 regions, $n = 2$ means 19 regions, and so on. Note that this parameter n computes the total number of regions that exist in the entire operational area (i.e. $R(n) = 3n^2 + 3n + 1$).

We evaluate the effect of the number of regions in the system on the overall cost ($\hat{C}_{\text{total}}$) given in Eq. (4) while varying other critical parameters (i.e., regional mobility rate σ, node density λ_p, and join rate to leave rate ratio $\lambda : \mu$) to test their effects. Table 4 lists the default parameter values of other parameters used in the case study. The wireless radio range (R) is selected as 200 m, as in IEEE 802.11, under which we observe a reasonable number of groups in response to group partition or merge events. The key size for the group key (k) or the intermediate key (v) is chosen to be 64 bits as commonly used for cryptographic keys. The beaconing intervals for the beacon messages by regional members and by leader members (T_{RB} and T_{LB}, respectively) are chosen to be 5 s and 2 s, respectively, to allow the system to quickly respond to node disconnection/failure events. The shorter interval is used for leader beaconing because of the importance of leader role. The size of a beaconing message (B_{alive}) is chosen to be only 32 bits since it is just a simple "I am alive" message. Finally, the size of a view update message (U_{view}) is chosen to be 500 bits so it is sufficiently large to contain information such as members' *id*s and their locations.

Fig. 11 shows optimal region sizes vs. mobility rate (σ). More specifically, Fig. 11a shows the impact of n on the total cost ($\hat{C}_{\text{total}}$) when varying σ, while Fig. 11b compares $\hat{C}_{\text{total}}$ between the baseline system (i.e. nonregion-based key management) vs. the system at the "optimum region size" (i.e. the optimal region size, 37 regions here, is used under region-based key management). We see that as the region size decreases (as n increases), $\hat{C}_{\text{total}}$ increases until it reaches the optimal point at $N_{\text{region}} = 37$ that would minimize $\hat{C}_{\text{total}}$, after which $\hat{C}_{\text{total}}$ increases again beyond that point.

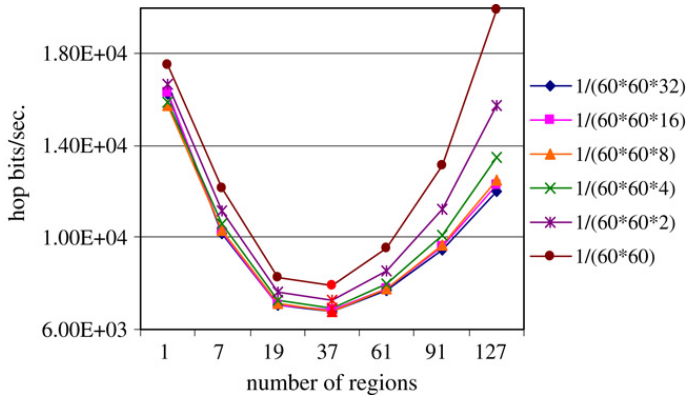


Fig. 11a. Overall cost ($\hat{C}_{\text{total}}$) vs. number of regions (N_{region}) as a function of mobility rate (σ).

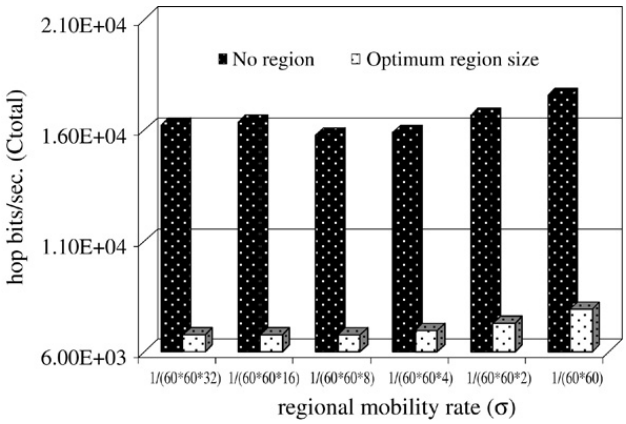


Fig. 11b. Overall cost ($\hat{C}_{\text{total}}$) in no region vs. in 37 regions as a function of mobility rate (σ).

Note that a higher N_{region} indicates there are fewer members in a region. The reason that an optimal N_{region} exists is that as N_{region} increases, the inter-regional overhead (i.e. updating and rekeying cost at a leader level) increases, while the intra-regional overhead (i.e. updating and rekeying cost at a regional level) 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 N_{region} reaches because of the increasing inter-regional overhead. As shown in Fig. 11b, the network traffic generated under the optimal region size is significantly lower than that under the no-region protocol.

Fig. 12a breaks down the overall cost ($\hat{C}_{\text{total}}$) into its constituents $\hat{C}_{\text{mobility}}$, $\hat{C}_{\text{join/leave}}$, $\hat{C}_{\text{beacon}}$, and $\hat{C}_{mp}$ as a function of N_{region} for the case in which $\sigma = 1/(60*60*4)$. We use this case to explain how an optimal $N_{\text{region}} = 37$, is obtained. The four cost components offset each other, and then the optimal point of total communication cost ($\hat{C}_{\text{total}}$) is determined at $N_{\text{region}} = 37$ by trading the intra-regional cost off for the inter-regional cost. As illustrated in Fig. 12b, while the "no region" protocol does not have $\hat{C}_{\text{mobility}}$, our protocol at the "optimum region size" has small $\hat{C}_{\text{mobility}}$. However, by having significant cost savings in three other cost components ($\hat{C}_{\text{beacon}}$, $\hat{C}_{\text{join/leave}}$, $\hat{C}_{mp}$) at the optimal region size, our protocol substantially improves the system performance. We also observed in both Figs. 12a and 12b that $\hat{C}_{\text{join/leave}}$ has more cost saving than $\hat{C}_{mp}$ and $\hat{C}_{\text{beacon}}$.

Fig. 13a shows the effect of node densities λ_p on $\hat{C}_{\text{total}}$. This case study varies λ_p from 150, 300, to 600. First, as λ_p increases, $\hat{C}_{\text{total}}$ increases because of the increased number of members in each region. Thus, increasing λ_p introduces a higher intra-regional cost for updating the regional membership information and rekeying the regional key. Here it should be noted that since the number of leaders remains the same under different λ_p, $\hat{C}_{\text{total}}$ increases as λ_p increases only due to an increased intra-regional overhead. Second, we observe that the optimal N_{region} shifts to the right as λ_p increases. That is, a smaller λ_p produces a smaller optimal N_{region}(e.g. optimal $N_{\text{region}} = 37$ at $\lambda_p = 150$ and 300) while a larger λ_p generates a larger optimal N_{region} (e.g. optimal $N_{\text{region}} = 61$ at $\lambda_p = 600$). This happens because the

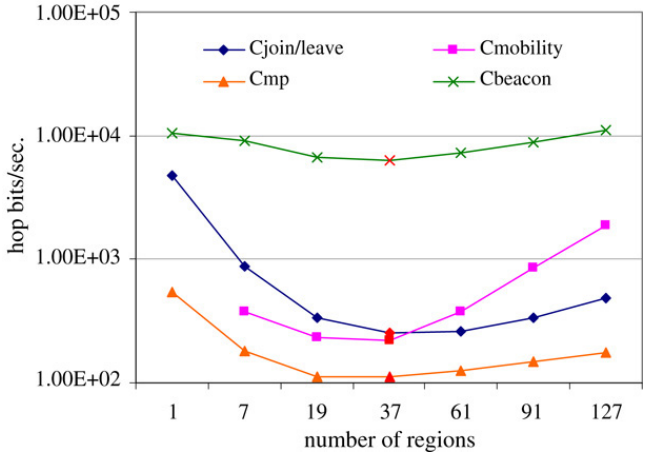
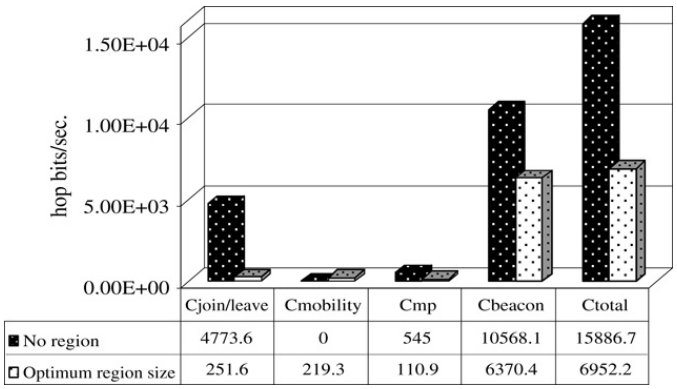


Fig. 12a. Breakdown of $\hat{C}_{\text{mobility}}$, $\hat{C}_{\text{join/leave}}$, $\hat{C}_{\text{beacon}}$, and $\hat{C}_{mp}$ vs. number of regions (N_{region}).



	Cjoin/leave	Cmobility	Cmp	Cbeacon	Ctotal
No region	4773.6	0	545	10568.1	15886.7
Optimum region size	251.6	219.3	110.9	6370.4	6952.2

Fig. 12b. $\hat{C}_{\text{mobility}}$, $\hat{C}_{\text{join/leave}}$, $\hat{C}_{\text{beacon}}$, and $\hat{C}_{mp}$ under no region vs. under optimal region size.

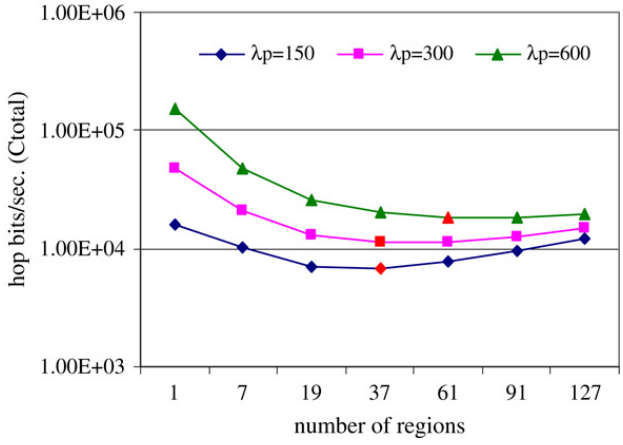


Fig. 13a. Overall cost ($\hat{C}_{\text{total}}$) vs. number of regions (N_{region}) as a function of node density (λ_p).

intra-regional cost always favours placing fewer members in a region and thus a smaller region size is favoured under high λ_p. Conversely, a large region size is preferred under low λ_p. Lastly, we notice that as N_{region} increases, $\hat{C}_{\text{total}}$ converges to almost the same point, e.g., $\hat{C}_{\text{total}}$ at $N_{\text{region}} = 127$. The reason is that in the extreme case where there are many regions, there is little intra-regional overhead and the inter-regional overhead dominates, thus causing $\hat{C}_{\text{total}}$ converged to the same value. Fig. 13b again shows that our protocol operating at the optimal region size produces

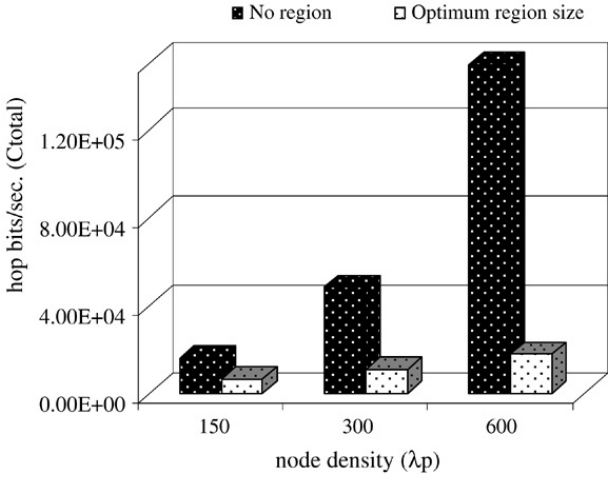


Fig. 13b. Overall cost ($\hat{C}_{\text{total}}$) under no region vs. under optimal region size as a function of node density (λ_p).

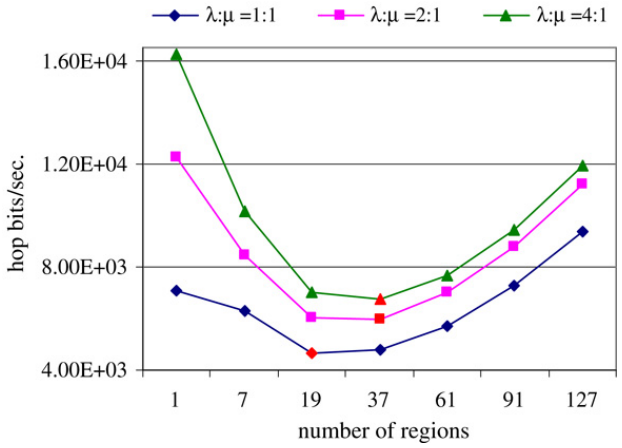


Fig. 14a. Overall cost ($\hat{C}_{\text{total}}$) vs. number of regions (N_{region}) as a function of the ratio of group join and group leave (λ: μ).

significant cost saving compared with the nonregion-based key management protocol. Noticeably, as λ_p increases, the cost saving is more pronounced.

Fig. 14a illustrates the impact of the ratio of join rate to leave rate (λ:μ) on $\hat{C}_{\text{total}}$. In this case study, the ratio of λ to μ varies from 1:1, 2:1, to 4:1 by increasing join rate (λ). As shown in Fig. 14a, as the absolute join and leave rates increase, $\hat{C}_{\text{total}}$ increases. Further, the optimal N_{region} (e.g. 37) increases as the join rate (e.g. λ: $\mu = 2$: 1 or 4:1) increases. This can be explained by the fact that since the join rate increases, the average number of nodes in a group increases, thus contributing to a higher intra-regional cost. As mentioned earlier, the intra-regional cost favours more regions to reduce the increased intra-regional cost. On the other hand, when a leave rate is relatively high (e.g. λ: $\mu = 1$: 1), there are fewer members in a group, thereby leading to a smaller intra-regional cost, while maintaining a constant inter-regional cost. Therefore, as the leave rate increases, fewer regions are favoured and a smaller optimal N_{region} is observed. Fig. 14b shows the cost saving when an optimal region size is used, compared with the case in which the nonregion-based key management protocol is used. Similar to the previous results (Figs. 12 and 13), it is strikingly noticeable that as a group has more members because of increased join rate, the cost saving of our protocol at the optimal region size becomes more significant compared with the no region protocol.

6.3. Simulation validation

We have conducted a simulation study to validate analytical results. The simulation program is implemented based on a discrete-event simulation language called *SMPL* [11].

We populate the MANET area based on a selected node density and the default parameter values listed in Table 4. For example, when $\lambda_p = 150$, we randomly place 150 nodes per km^2, thus having a total of 480 nodes approximately

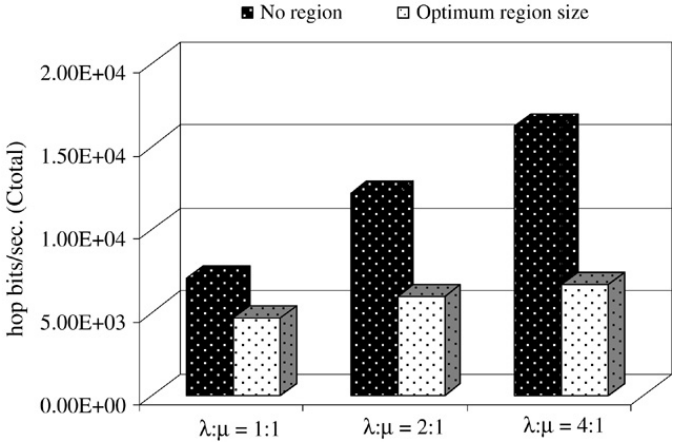


Fig. 14b. Overall cost ($\hat{C}_{\text{total}}$) under no region vs. under optimal region size as a function of the ratio of group join and group leave (λ:μ).

in the π km^2 operational area in our simulation. In this case, the initial number of member nodes, i.e. $N\lambda/(\lambda+\mu)$, is 392 and they are scattered in the operational area. All nodes can be connected through multiple hops using a per-hop wireless radio range (R) of 200 m. Multiple groups may be observed in the operational area. Each node in its lifecycle could generate six events, namely, GROUP JOIN, GROUP LEAVE, BEACON, MOBILITY (i.e. a node's regional boundary crossing), GROUP MERGE, and GROUP PARTITION. For the mobility event, we use *random waypoint mobility (RWM)* [12] to model the movement of a node with mobility rate of σ by setting the pause time as zero and the speed as

$$S(\sigma) = \frac{2r}{\text{expntl}(1/\sigma)} \tag{32}$$

Here r is the MANET area radius, and expntl(x) returns a random number based on an exponential distribution with mean x. For MOBILITY events generated by a node, the time at which a boundary crossing will occur is calculated based on the current speed and direction of the node, and then a MOBILITY event is scheduled accordingly. For GROUP JOIN and GROUP LEAVE events, we assume that the inter-arrival times are exponentially distributed with the rates of λ and μ respectively. Thus, GROUP JOIN and GROUP LEAVE events are scheduled based on random values generated by expntl($1/\lambda$) and expntl($1/\mu$). Lastly, BEACON events are scheduled periodically.

The whole MANET area is divided into equal-sized hexagons based on the hexagonal network coverage model with parameter n. Nodes are confined within the MANET area but otherwise can move freely within that area based on the *RWM* model. The simulation keeps track of the location of each node. Thus, at any given time, it knows which region a member belongs to. Whenever an event that affects a regional view occurs, the regional view information is updated. The simulation is event-driven and the cost associated with each event is calculated based on the knowledge of exact locations of nodes in calculating the number of hops, i.e. H_{region} and $H_{\text{leader},i}$. We used a technique called *batch mean analysis* [11] to assure the statistical significance of our simulation results in terms of the cost measurements collected (each called an *observation*). The simulation period is divided into 10 batches with each batch consisting of 200,000 observations. Tests are performed to guarantee that the number of observations is large enough so that the batch means are approximately independent and normally distributed. The results are obtained with a confidence interval (CI) of 95% and accuracy level of 10% from the true mean. Further, in order to remove the initial transient (warm-up) problem, the first batch discards the first 200 sample values. Also to compare values of different system configurations, the *variance reduction technique (VRT)* [11] of *common random numbers (CRN)* has been used in this case study. Default parameter values as listed in Table 4 are used to set up the simulation environment.

Fig. 15 shows the results obtained from our simulation under the same conditions of Fig. 13. These curves look remarkably similar to those obtained analytically exhibiting the same trend despite the fact that one set is calculated analytically while the other set is obtained through simulation by averaging the statistical data collected through batch mean analysis. The very slight difference between analytical results and simulation results is due to the way we calculate the number of hops for computing intra- and inter-regional costs and the way we use *RWM* to approximate the per-node mobility rate. In the simulation, we keep track of the location of each node and use exact locations of

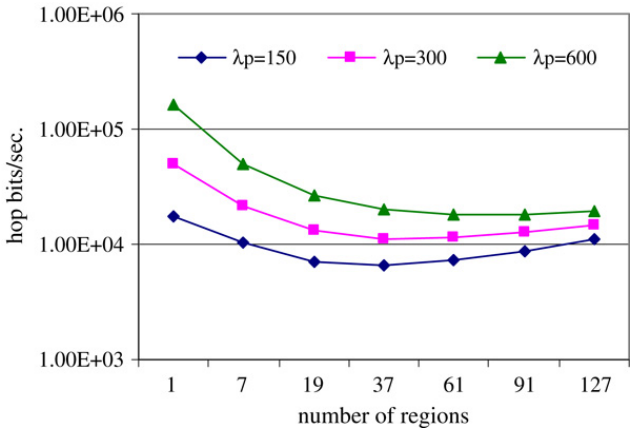


Fig. 15. Simulation results: overall cost ($\hat{C}_{\text{total}}$) vs. number of regions (N_{region}) as a function of node density (λ_p).

nodes in the system to calculate the number of hops separating any two nodes when calculating the average $C^{\text{intra}}_{\text{rekey}}$ and $C^{\text{inter}}_{\text{rekey},i}$, while in the analysis, we use Eqs. (14), (15), (17) and (18) to compute the number of hops. Nevertheless, the average cost obtained from simulation along with the trend exhibited as the number of regions increases is remarkably close to that obtained analytically.

7. Conclusion

In this paper, we have proposed and analysed a scalable and efficient region-based secure group key management protocol to support secure group communications in mobile ad hoc networks. The region-based hierarchical key proposed not only reduces network communication costs, but also provides robust security properties. By using GDH as an example key agreement protocol for group key generation and rekeying at the intra-regional and inter-regional (leader) levels, we discovered that there exists an optimal region size that would minimize the overall network traffic when given a set of parameter values characterizing the operational condition. The existence of the optimal region size is a tradeoff between inter-regional and intra-regional overheads, and it is sensitive to identified system parameters, such as the node density, node mobility rate, and the group join/leave rate in our case study. In the future, we plan to extend the protocol to consider energy consumption issues. Lastly, the protocol coupled with authentication can only deal with outsider attacks. We plan to extend the protocol to consider insider attacks and intrusion detection.

Acknowledgement

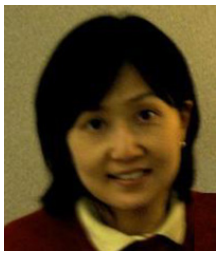
This research work was partly supported by a National Science Foundation IGERT grant #9987586.

References

[1] Y. Amir, C. Nita-Rotaru, J.L. Schultz, J. Stanton, G. Tsudik, Secure spread: An integrated architecture for secure group communication, IEEE Transactions on Dependable and Secure Computing 2 (3) (2005) 248–261.
[2] Y. Amir, Y. Kim, C. Nita-Rotaru, G. Tsudik, On the performance of group key agreement protocols, ACM Transactions on Information and System Security 7 (3) (2004) 457–488.
[3] Y. Amir, Y. Kim, C. Nita-Rotaru, J.L. Schultz, J. Stanton, G. Tsudik, Secure group communication using robust contributory key agreement, IEEE Transactions on Parallel and Distributed Systems 15 (5) (2004) 468–480.
[4] Y. Kim, A. Perrig, G. Tsudik, Tree-based group key agreement, ACM Transactions on Information and System Security 7 (1) (2004) 60–96.
[5] Y. Kim, A. Perrig, G. Tsudik, Communication-efficient group key agreement, in: Proc. IFIP TC11 16th Annual Working Conf. on Information Security, June 2001, pp. 229–244.
[6] X.S. Li, Y.R. Yang, M.G. Gouda, S.S. Lam, Batch rekeying for secure group communications, in: Proc. 10th Int'l World Wide Web Conf. on World Wide Web (WWW10), Hong Kong, May 2001, pp. 525–534.
[7] O. Rodeh, K. Birman, D. Dolev, Using AVL trees for fault tolerant group key management, International Journal on Information Security 1 (2) (2001) 84–99.
[8] 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.

[9] J.W. Wilson, I.R. Chen, Performance characteristics of location-based group membership and data consistency algorithms in mobile ad hoc networks, International Journal of Wireless and Mobile Computing 1 (8) (2005) 1–12.
[10] 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.
[11] M.H. MacDougall, Simulating Computer Systems, MIT Press, Cambridge, MA, USA, 1987.
[12] T. Camp, J. Boleng, V. Davies, A survey of mobility models for ad hoc network research, Wireless Communication & Mobile Computing 2 (5) (2002) 483–502.
[13] T. Hardjono, B. Cain, I. Monga, Intra-domain group key management protocol, Internet Draft (1998).
[14] S. Rafaeli, D. Hutchison, A survey of key management for secure group communication, ACM Computing Surveys (CSUR) 35 (3) (2003) 309–329.
[15] M. Younis, K. Ghumman, M. Eltoweissy, Location-aware combinatorial key management scheme for clustered sensor networks, IEEE Transactions on Parallel and Distributed Systems 17 (8) (2006) 865–882.
[16] C. Duma, N. Shahmehri, P. Lambrix, A hybrid key tree scheme for multicast to balance security and efficiency requirements, in: Proc. 12th Int'l Workshop Enabling Technologies: Infrastructure for Collaborative Enterprises, June 2003, pp. 208–213.
[17] Jason H. Li, Renato Levy, Miao Yu, Bobby Bhattacharjee, A scalable key management and clustering scheme for ad hoc networks, in: Proc. 1st ACM Int'l Conf. on Scalable Information Systems, vol. 152, no. 28, Hong Kong, May 2006.
[18] 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.
[19] L. Dondeti, S. Mukherjee, A. Samal, Scalable secure one-to-many group communication using dual encryption, Computer Communications 23 (17) (2000) 1681–1701.
[20] S. Mittra, Iolus: A framework for scalable secure multicasting, in: Proc. ACM SIGCOMM'97, vol. 27, no. 4, Cannes France, September 1997, pp. 277–288.
[21] S. Banerjee, B. Bhattacharjee, Scalable secure group communication over IPMulticast, JSAC Special Issue on Network Support for Group Communication 20 (8) (2002) 261–271.
[22] M. Steiner, G. Tsudik, M. Waidner, Key agreement in dynamic peer groups, IEEE Transactions on Parallel and Distributed Systems 11 (8) (2000) 769–780.
[23] M. Eltoweissy, M. Moharrum, R. Mukkamala, Dynamic key management in sensor networks, IEEE Communications Magazine 44 (4) (2006) 122–130.
[24] C. Becker, U. Wille, Communication complexity of group key distribution, in: Proc. 5th ACM Conf. on Computer and Communications Security, San Francisco, CA, Nov. 1998, pp. 1–6.
[25] C.K. Wong, M. Gouda, S.S. Lam, Secure group communications using key graphs, IEEE/ACM Transactions on Networking 8 (1) (2000) 16–30.
[26] O. Rodeh, K. Birman, D. Dolev, Optimized group rekey for group communication systems, in: Proc. of ISOC Network and Distributed System Security Symposium, San Diego, CA, February 2000.
[27] L. Dondeti, S. Mukherjee, A. Samal, A distributed group key management scheme for security many-to-many communication, Technical Report, PINTL-TR-207-99, Department of Computer Science, University of Maryland, 1999.
[28] Y. Kim, A. Perrig, G. Tsudik, Tree-based group key agreement, ACM Transactions on Information and System Security 7 (1) (2004) 60–96.
[29] A. Perrig, J.D. Tygar, Secure Broadcast Communication in Wired and Wireless Networks, Kluwer Academic Publishers, New York, USA, 2002.
[30] H. Yang, H. Luo, F. Ye, S.W. Lu, L. Zhang, Security in mobile ad hoc networks: Challenges and solutions, IEEE Wireless Communications 11 (1) (2004) 38–47.
[31] I. Ingemarsson, D.T. Tang, C.K. Wong, A conference key distribution system, IEEE Transactions on Information Theory 28 (5) (1982) 714–720.
[32] M. Burmester, Y. Desmedt, A secure and efficient conference key distribution system, in: Advances in Cryptology: EUROCRYPT94, vol. 950, 1994, pp. 275–286.
[33] D.A. McGrew, A.T. Sherman, Key establishment in large dynamic groups using one-way function trees, Technical Report, No. 0755, TIS Labs at Network Associates, Inc., 1998.
[34] A. Balasubramanian, S. Mishra, R. Sridhar, Analysis of a hybrid key management solution for ad hoc networks in: Proc. 2005 IEEE Wireless Communications and Networking Conf., vol. 4, 2005, pp. 2082–2087.
[35] Kyung Hyune Rhee, Young Ho Park, Tsudik Gene, An architecture for key management in hierarchical mobile ad-hoc networks, Journal of Communications and Networks 6 (2) (2004) 156–162.
[36] M. Bechler, H.-J. Hof, D. Kraft, F. Pahlke, L. Wolf, A cluster-based security architecture for ad hoc networks, in: Proc. 23rd IEEE INFOCOM, vol. 4, March 2004, pp. 2393–2403.
[37] S. Basagni, Distributed clustering for ad hoc networks, in: 1999 Int'l Symposium on Parallel Architectures, Algorithms and Networks, IEEE Computer Society, Australia, 23–25 June 1999, pp. 310–315.
[38] Y. Wang, X. Li, O. Frieder, Efficient hybrid key agreement protocol for wireless ad hoc networks, in: Proc. 11th Int'l Conf. on Computer Communications and Networks, Oct. 2002, pp. 404–409.
[39] L. Lazos, R. Poovendran, Energy-aware secure multicast group communication in mobile networks using geographic location information, in: Proc. IEEE Int'l Conf. on Acoustics Speech and Signal Processing, vol. 4, Hong Kong, April 2003, pp. 201–204.
[40] L. Lazos, R. Poovendran, Location-aware secure wireless multicast in ad hoc networks under heterogeneous pathloss, UWEETR-2003-0012, UWEE Technical Report Series, 2003.

[41] I.R. Chen, J.H. Cho, D.C. Wang, Performance characteristics of region-based group key management in mobile ad hoc networks, in: 1st IEEE Inter'l Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing Taichung, Taiwan, June 2006, pp. 411–419.
[42] P.C. Lee, C.S. Lui, K.Y. Yau, Distributed collaborative key agreement and authentication protocols for dynamic peer groups, IEEE/ACM Transactions on Networking 14 (2) (2006) 263–276.
[43] E. Jung, A.X. Liu, M. Gouda, Key bundles and parcels: Secure communication in many groups, Computer Networks 50 (11) (2006) 1781–1798.



Jin-Hee Cho received the B.A. degree from Ewha Womans University in Seoul, Korea in 1997, and the M.S. degree in Computer Science from Virginia Polytechnic Institute and State University, USA, in 2004. Since Fall 2004, she has been pursuing her Ph.D. degree in the Department of Computer Science at Virginia Tech, where she is a Graduate Research Assistant in the Mobile Computing Lab. Her research interests include wireless mobile networks, mobile ad hoc networks, sensor networks, secure group communication, network security, and intrusion detection systems.

Ing-Ray Chen received the B.S. degree from the National Taiwan University, Taipei, Taiwan, and the M.S. and Ph.D. degrees in computer science from the University of Houston. He is a professor in the Department of Computer Science at Virginia Tech. His research interests include mobile computing, pervasive computing, multimedia, distributed systems, real-time intelligent systems, and reliability and performance analysis. Dr. Chen has served as program chair and program committee member for numerous international conferences. Dr. Chen currently serves as an editor for *Wireless Personal Communications, The Computer Journal, Wireless Communications and Mobile Computing, and International Journal on Artificial Intelligence Tools*. He is a member of the IEEE/CS and ACM.

Ding-Chau Wang received the B.S. degree from Tung-Hai University, Taichung, Taiwan, and the M.S. and Ph.D. degrees in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan. He is currently an assistant professor in the Department of Information Management at the Southern Taiwan University of Technology. His research interests include distributed systems, mobile computing, and performance analysis.