Performance Evaluation 68 (2011) 58–75

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

Performance analysis of hierarchical group key management integrated with adaptive intrusion detection in mobile ad hoc networks

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

[a] Computational and Information Sciences Directorate, US Army Research Laboratory, United States

[b] Department of Computer Science, Virginia Tech, United States

ARTICLE INFO

Article history:
Received 7 September 2009
Received in revised form 13 July 2010
Accepted 19 September 2010
Available online 26 September 2010

Keywords:
Mobile ad hoc networks
Intrusion detection
Group communication systems
Group key management
Region-voting-based IDS
Host-based IDS
Stochastic Petri net
Performance analysis

ABSTRACT

We develop a mathematical model to quantitatively analyze a scalable region-based hierarchical group key management protocol integrated with intrusion detection to deal with both outsider and insider security attacks for group communication systems (GCSs) in mobile ad hoc networks (MANETs). Our proposed adaptive intrusion detection technique is based on majority voting by nodes in a geographical region to cope with collusion of compromised nodes, with each node preloaded with anomaly-based or misuse-based intrusion detection techniques to diagnose compromised nodes in the same region. When given a set of parameter values characterizing operational and environmental conditions, we identify the optimal intrusion detection rate and the optimal regional area size under which the mean time to security failure of the system is maximized and/or the total communication cost is minimized for GCSs in MANET environments. The tradeoff analysis in performance versus security is useful in identifying and dynamically applying optimal settings to maximize the system lifetime for scalable mobile group applications while satisfying application-specific performance requirements.

Published by Elsevier B.V.

1. Introduction

Many mobile applications in wireless networks such as military battlefields, emergency response, mobile commerce, online gaming, and collaborative work are based on the notion of group communications. Designing security protocols for secure group communication systems (GCSs) in mobile ad hoc networks (MANETs) faces many technical challenges including resource-constrained environments (e.g., bandwidth, memory size, battery life, and computational power), openness to eavesdropping and security threats, unreliable communication, no infrastructure support, and rapid changes in network topology due to user mobility which could cause group merge/partition events to occur dynamically [1].

In this paper, we are concerned with dynamic GCSs in MANETs where mobile nodes cooperate in a group setting to accomplish assigned mission tasks as in military battlefield situations. A GCS initially may contain one mobile group. Later a mobile group may be split into two because of node mobility and failure. Two mobile groups may merge into one when connectivity is resumed. Our notion of a mobile group is connectivity-oriented, i.e., nodes are in a mobile group as long as they are connected.

The GCS is mission-oriented in that all mobile groups regardless of their size will execute the assigned mission throughout their lifetime. Thus, the primary goal of a mission-oriented GCS is to prolong its system lifetime in the presence of security attacks to increase the success probability of mission execution. The secondary goal is to satisfy application-specific performance requirements in terms of timeliness, throughput, delay, or traffic capacity. This paper aims to solve the problem

* Corresponding author. Tel.: +1 703 538 8376; fax: +1 703 538 8348.
E-mail addresses: jinhee.cho@us.army.mil (J.-H. Cho), irchen@vt.edu (I.-R. Chen).

0166-5316/$ – see front matter. Published by Elsevier B.V.
doi:10.1016/j.peva.2010.09.005

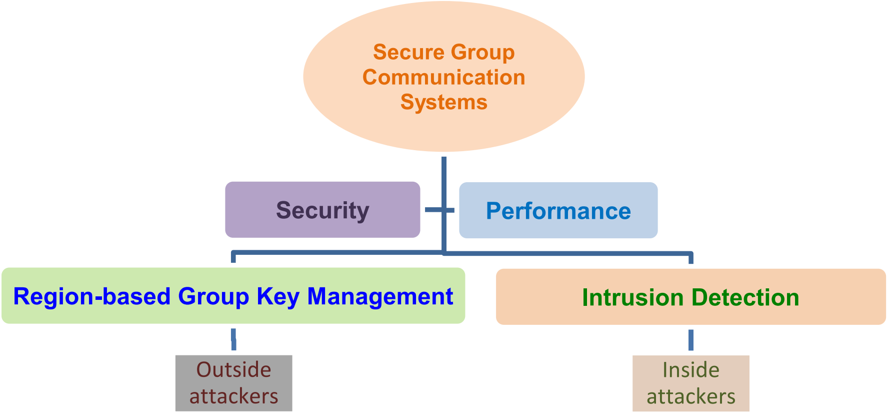


Fig. 1. Integration of region-based group key management for dealing with outsider attacks with intrusion detection for dealing with insider attacks for a GCS in MANETs.

of prolonging the system lifetime of such a GCS in the presence of security attacks while satisfying system performance requirements.

To deal with *outsider attacks*, a cost-effective way is to use a symmetric key, called the *group key*, shared by group members. The group key is employed to encrypt messages sent by a member to the group. Only members of the group with the group key are capable of decrypting the messages. Thus, the group key protects group communication information shared by legitimate members. Since there is no infrastructure support in MANETs, key management must be performed in a fully distributed manner. This creates extra system overheads whenever the group key is "rekeyed" because of a group member leave/join/eviction event. For scalability and performance reasons, we advocate the use of region-based hierarchical group key management [2] for generating and distributing the group key.

To deal with *insider attacks*, intrusion detection system (IDS) techniques [3] have been proposed. IDS can be employed to detect compromised nodes and to evict such compromised nodes with the goal of high survivability and availability to prolong the system lifetime. In MANETs, distributed IDS must be employed because it is not practical to use a centralized IDS server which would be a single point of failure. We advocate the use of voting-based IDS [4] to cope with collusion of compromised nodes for survivability.

In the literature, group key management and IDS techniques for MANETs have been studied separately to deal with outside attackers and inside attackers, respectively. However, both are needed in a GCS to ensure high system survivability and security. As a result, group key management and IDS are often deployed separately, resulting in high management overhead and degraded performance. In this paper, we aim to demonstrate the advantages of integrated management over separate management. Specifically, we propose to integrate region-based hierarchical group key management [2] with a new region-voting-based IDS scheme to result in a highly survivable and efficient GCS. We use the *mean time to security failure* (*MTTSF*) as the security metric to measure the lifetime of a GCS system under security attacks. We use the control message traffic generated for secure group communication as the performance metric. We analyze the tradeoff of security versus performance properties of a GCS system by means of a mathematical model and identify optimal settings such that the system *MTTSF* is maximized while the total control message traffic incurred is minimized in the GCS. Fig. 1 shows the structure of our integrated protocol to deal with both outsider and insider attacks while satisfying security and performance requirements for GCSs in MANETs. We shall demonstrate that integrated management yields higher *MTTSF* while producing lower control message traffic compared with separate deployment of key management and IDS techniques in the literature.

The contributions of the paper are as follows. First, we consider an integrated design to cope with both outsider and insider security attacks, i.e., region-based group key management for dealing with outsider attacks and region-voting-based IDS for dealing with insider attacks, with the goal to achieve secure GCSs in MANETs. This is the first work that investigates the interplay between group key management and IDS and analyzes their interwoven effect of how one affects the optimal setting of the other for maximizing the mean lifetime of the system. Second, we analyze the tradeoff of security versus performance properties of GCSs as a result of integrating IDS with hierarchical group key management for scalability, survivability, and efficiency. Third, we introduce region-voting-based IDS to cope with collusion of compromised nodes to prolong the system *MTTSF* with a randomness design in the vote-participants and region-leader selection process to mitigate the problem of a single point of attack/failure. Lastly, we develop a mathematical model based on stochastic Petri nets (SPN) [5] to quantitatively identify optimal settings in terms of the optimal regional area size and the intrusion detection interval that would maximize the system lifetime (in terms of *MTTSF*) while satisfying performance requirements (i.e., total communication cost per time unit). The SPN model developed is capable of dealing with a large number of states and the underlying semi-Markov chain generated can accommodate any general time distribution for modeling the event occurrence duration, thus allowing periodic events and non-exponential-time events to be easily analyzed. A GCS can monitor the operational and environmental conditions characterized by a set of model parameter values at runtime and apply the optimal settings identified in this paper at static time by means of a table look-up operation so as to dynamically maximize the system *MTTSF* and satisfy imposed performance requirements.

Table 1
New contributions to region-based hierarchical group key management.

	Region-based group key management [2]	This work
Insider attacks	No	Yes
Outsider attacks	Yes	Yes
Optimization design tradeoffs	Performance tradeoff	Performance versus security tradeoffs
Optimal settings	Optimal regional area size	Optimal regional area size and optimal intrusion detection interval
Metrics	Overall communication cost	Overall communication cost, *MTTSF*
Integration of group key management with IDS	No	Yes
Events considered	Join, leave, group merge, group partition, mobility, and beaconing	Join, leave, group merge, group partition, mobility, beaconing, group communication, intrusion detection, and node eviction

As this paper integrates region-based hierarchical group key management [2] with region-voting-based IDS to deal with both outsider and insider attacks, we summarize this paper's new contributions with respect to our prior work in region-based hierarchical group key management in Table 1.

The rest of this paper is organized as follows. Section 2 surveys related work in dealing with insider and outsider attacks for secure GCSs in MANETs. Section 3 describes our proposed integrated hierarchical group key management and region-voting-based IDS protocol. Section 4 describes the system model covering the assumptions used, security failure conditions defined, performance and security metrics used, and attack model. Section 5 develops a mathematical model based on SPN and explains how we model group merge/partition events and how we calculate security and performance metrics from the mathematical model developed. Section 6 shows analytical results obtained with physical interpretations given. Section 7 discusses the applicability of the proposed integrated protocol and summarizes the paper with future work outlined.

2. Related work

Several hierarchical group key management protocols have been proposed in the literature for scalability reasons. IGKMP [7,6] divides a group into several subgroups to enhance scalability and applies several rekeying algorithms to preserve secrecy properties as members move within the hierarchy. However, it is not suitable for MANETs where nodes are mobile and wireless communication is often unreliable. Rafaeli et al. [8] proposed *Hydra*, Dondeti et al. [9] proposed DEP, and Mittra et al. [10] proposed *Iolus* for secure multicasting communication. However, all are based on the use of static subgroup controllers in the system and are also not suitable for MANETs. In [11], 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. However, HKT is designed for wired networks. Hierarchical group key management protocols developed for MANETs include [12–19]. Rhee et al. [18] employed a two-layer hierarchical key management structure for secure group communications for unmanned aerial vehicles (UAVs). They considered the use of a stationary super-node as a cluster head. Bechler et al. [15] proposed an efficient distributed key management based on hierarchical clustering. However, no optimal setting was identified to maximize system performance. Lazos et al. [16] considered a hierarchical key management structure for energy-aware secure multicast group communication in MANETs based on geographic routing. They assumed a fixed cluster size without identifying the optimal cluster size to maximize system performance. Furthermore, they only considered group join/leave events without considering group partition/merge events due to node mobility. In [12,13,17,20], the optimal cluster or group size to minimize rekeying cost was studied. The analysis, however, is not for mission-oriented GCSs in MANETs.

Very recently, we investigated a region-based group key management protocol [2] to improve the system performance for group key generation and distribution for GCSs in MANETs. Their protocol breaks the operational area into *regions* to reduce the group key management overhead and to make the protocol scalable to a large number of nodes in a group. An optimal regional area size was identified based on the tradeoff between inter-regional and intra-regional communication cost. However, it deals with outsider attacks only with the focus on communication cost minimization. This paper extends [2] by integrating the hierarchical group key management protocol with region-voting-based IDS in order to deal with both insider and outsider attacks, with the goal to achieve not only high scalability and efficiency, but also high survivability for GCSs in MANETs.

Recent research efforts on IDS protocols developed for MANETs focus on cluster-based IDS for scalability. A major issue is that the cluster head is a single point of failure. The main idea behind cluster-based IDS is that instead of performing host-based IDS at each node, a cluster head (CH) is selected to collect security-related information from nodes in a cluster and determines if intrusion has occurred. Non-overlapping zone-based IDS was proposed in [21,22] for MANETs and proven to be effective in intrusion detection. Marti et al. [23] developed a *watchdog* mechanism for identifying misbehaving nodes based on dynamic behaviors and developed a *pathrater* algorithm for routing around misbehaving nodes for MANETs. Debar et al. [24] suggested aggregation and correlation of IDS alerts to reduce communication/computational overhead caused by performing IDS. Hierarchical IDS was proposed in [25–27] to realize distributed anomaly-based IDS in MANETs. However, the issues of extra latency and energy consumption are not addressed. The assumptions that the CH is tamper-resistant and the CH selection process will not be interrupted by attackers are also questionable. We advocate region-voting-based IDS by which multiple nodes in a region participate in the eviction process of a target node and perform a distributed majority voting to determine if the target node is to be evicted. This eliminates a single point of failure in cluster-based IDS protocols.

Little work addresses both security and performance issues of IDS. Stern et al. [28] proposed data reduction techniques to reduce the communication cost in their IDS design. Subhadrabandhu et al. [29–31] studied the tradeoff between energy, computational, and communication resource consumption versus IDS accuracy based on distributed IDS. Our work also concerns the effect of IDS on security and performance properties of the system since we aim to determine the best IDS operational settings to prolong the system lifetime against security attacks while satisfying the performance requirement. However, our work differs from these prior works in that we integrate region-voting-IDS with hierarchical key management to deal with both insider and outsider attacks and our integrated protocol specifically deals with secure GCSs in MANET environments.

Our work has its root in model-based quantitative analysis [32]. In the literature, not much work has been done in extending model-based quantitative analysis to security analysis. Zhang et al. [6] analyzed several group rekeying algorithms in wireless environments and evaluated their performance characteristics. Dacier et al. [33] proposed a system model using a privilege graph demonstrating operational security vulnerabilities and transformed the privilege graph into a *Markov chain* based on all possible successful attack scenarios. Jonsson et al. [34] presented a quantitative *Markov* model of attacker behaviors by proposing multiple phases, such as learning, standard attack, and innovative attack. Popstojanova et al. [35] presented a state transition model to describe dynamic behaviors of intrusion tolerance systems including a framework defining the vulnerability and the threat set. Madan et al. [36,37] employed a *Semi-Markov* Process (SMP) model to evaluate security attributes of an intrusion-tolerant system. A steady-state analysis has been used to obtain dependability measures such as availability. A transient analysis with absorbing states has been used to obtain security measures such as *MTTSF* similar to the computation of the *mean time to failure* (*MTTF*) in reliability analysis. Stevens et al. [38] also proposed a networked intrusion tolerant information system using a model-based validation technique, the so called probabilistic modeling. They used two security metrics: the *mean time to discovery* (*MTTD*) refers to the mean time between successive discoveries of unknown vulnerabilities and the mean time to exploit (*MTTE*) refers to the mean time between successive exploitations of a known vulnerability. Wang et al. [39] utilized a higher-level formalism based on *SPN* for security analysis of intrusion tolerant systems. Recently, Leversage and James [40] suggested a security metric to intelligently compare systems and to make corporate security decisions. They proposed a *mean time-to-compromise (MTTC)* metric to measure the time needed for an attacker to successfully disrupt a target system. Most previous works cited above, however, often only focused on security measures without considering the impact of deploying security protocols on the performance of the system. In this paper, we develop model-based analysis techniques that consider both security and performance aspects with the objective to identify operational settings under which both security and performance requirements can be best satisfied.

To the best of our knowledge, no existing work considers the integration of group key management (to deal with outsider attacks) with intrusion detection mechanisms (to deal with insider attacks) for high scalable, reconfigurable, and survivable GCSs in MANETs. Our work is the first that considers these two mechanisms and their interplay with analysis techniques being developed to determine the best setting to execute the integrated protocol to satisfy both the security and performance requirements for GCSs in MANETs.

3. Preliminaries

3.1. IDS protocol

To effectively integrate IDS with region-based hierarchical group key management and investigate their interplay, we develop a new *region-voting-based* IDS protocol borrowing the concepts of distributed revocation based on majority voting in sensor networks [41] and voting-based IDS design in MANETs [4]. The protocol requires each node to preinstall host-based IDS to evaluate its neighbors based on evidences collected, mostly route-related and traffic-related information [26]. Route-related information may include node velocity, new and stale routes, routes added by overhearing, repaired routes, route changes, and the average route length. Traffic-related information may include packet type (e.g., data, route request, route reply), flow direction (e.g., received, sent, forwarded, dropped), sampling interval, and statistics measures (e.g., the count and standard deviation of inter-packet intervals) [26,42]. Anomalies of route-related and traffic-related information can show evidence of malicious attacks. Each node can also *actively* collect information such as observing if a packet sent to a neighbor is not forwarded as requested. For example, in the MAC layer, it may include the total number of channel requests, the total number of nodes making channel requests, and the largest, the mean, and the smallest of all requests. In the application layer, it may include the total number of requests made to a service, the number of different services requested, the average duration of a service, the number of nodes that have requested any service, and the total number of service errors.

The host-based IDS preinstalled in each node can be characterized by two parameters, namely, the *per-node* false negative probability (*p1*) and *per-node* false positive probability (*p2*). The host-based IDS can use any general IDS technique characterized by *p1* and *p2* such as misuse detection (also called signature-based detection) or anomaly detection [29]. Under region-voting-based IDS, compromised nodes are detected based on majority voting. Periodically, a target node would be evaluated by *m* vote-participants dynamically and randomly selected out of group members in the same geographical region (see Section 3.2 below for the meaning of a "region" under region-based hierarchical group key management).

Our proposed region-voting-based IDS protocol performs its detection and eviction function periodically. In a detection interval, each node would be evaluated by *m* vote-participants where *m* is a system parameter whose effect will be analyzed in the paper. All nodes are loosely synchronized and votes are tallied only during the current detection time interval. The *m* vote-participants are randomly selected from members within the same region as follows. Since members within the same region know each other's *id* and location, each node will utilize a preinstalled hash function *R(x)* that takes in the $id + location$ of the target node concatenated with the $id + location$ of every other node as the hash key. The *m* nodes that yield the highest hash values will be selected as the *m* vote-participants voting for or against the target node. The main benefit of using a randomization function to determine *m* vote-participants is that it does require any communication between nodes. Because there is no vote-coordinator, it avoids a single point of attack/failure. Each vote-participant node selected will independently vote for or against the target node by disseminating its vote to all other regional members. Vote authenticity is achieved via preloaded public keys of all other group members. By this way, all regional members know who *m* vote-participants are, and, based on votes received, can determine whether or not a target node is considered compromised and needs to be evicted for security reasons.

Since each node has a unique *id*, all group members know which target node is being evaluated. If the majority decided to vote against the target node, then the target node would be evicted from the system. This adds intrusion tolerance to tolerate collusion of compromised nodes in MANETs. We characterize region-voting-based IDS by two parameters, namely, false negative probability (P_{fn}) and false positive probability (P_{fp}). Later we will derive a formula to assess these two parameters based on (a) the *per-node* false negative and positive probabilities (*p1* and *p2*); (b) the number of vote-participants, *m*, selected to vote for or against a target node; and (c) an estimate of the current number of compromised nodes which may collude with the objective to disrupt the service of the system. Since *m* nodes are selected to vote, if the majority of *m* voting-participants (i.e., $>\lceil m/2\rceil$) casts negative votes against a target node, the target node is considered compromised and will be evicted from the system. Here *p1* and *p2* are probabilities for characterizing how often a good node will make a false alarm. When a good node evaluates a bad node, it will fail to identify it as a bad node with probably *p1*; when a good node evaluates another good node, it will misdiagnose it as a bad node with probability *p2*. A false alarm from one good node is not harmful as long as there are more than $\lceil m/2\rceil$ other good nodes correctly evaluating the target node in the voting process. In the special case in which there is only a single region, all nodes in the group are candidates as vote-participants against a target node. The selection of *m* vote-participants from member nodes from a geographical region of different sizes affects the security and performance aspects of region-voting-based IDS for which we aim to study its effect in this paper.

3.2. Region-based group key management protocol

In this paper, we extend region-based group key management [2] to include intrusion detection events to tolerate insider attacks. The region-based group key management scheme divides a group into region-based subgroups based on *decentralized* key management principles. This protocol requires each group member to be equipped with GPS to know its location as it moves across regions. When a member crosses a regional boundary, it changes its subgroup "regional" membership although it is still a member of the group. For secure group communications [43], all group members share a secret group key, K_G. All subgroup members in region *i* share a secret regional key, K_{Ri}. In addition, each region has a leader and all leaders in the system share a *leader secret key* (K_{RL}) for efficiency purposes. In summary, there are three keys for hierarchical group key management: *leader key* (K_{RL}), *regional key* (K_R), and *group key* (K_G). These keys are rekeyed properly, in part or in whole, as events happen in the system, including group join/leave/eviction, node failure, regional boundary cross, and group merge/partition. In this paper, we consider additional events associated with IDS activities, including intrusion detection timer events and eviction events. In addition to maintain secrecy, our region-based key management scheme also maintains membership consistency through three *membership views*: (a) *Regional View (RV)* contains regional membership information including regional (or subgroup) members' *ids* and their location information, (b) *Leader View (LV)* contains leaders' *ids* and their location information, and (c) *Group View (GV)* contains group membership information that includes members' *ids* and their location information [2,44].

For robustness and distributed control, a contributory key agreement (*CKA*) protocol based on Group Diffie–Hellman (*GDH*) [45] is used to generate and manage a shared secret key. In particular, we use GDH.3 (the optimized version of GDH) allowing the use of fixed-sized messages and only a constant number of exponentiation operations executed by each participant. With these features, GDH.3 has been proposed to be used for rekeying by mobile devices with low computational capabilities [45]. If an inside attacker interrupts the rekeying process, then a timeout event will be triggered to restart the rekeying process. A smart inside attacker, however, may decide to allow the rekeying operation to be executed to completion so that it can learn of the new secret group key with which to obtain secret group information and cause the system to fail. If it continues to interrupt rekeying, it may eventually be caught by IDS and evicted from the system before it causes the system to fail.

A regional leader will be selected randomly among regional members so that the adversary will not have a specific target to launch their attacks. We add randomness to the leader selection process by introducing another hashing function *H(x)* that takes in the *id* of a node concatenated with the current *location* of the node as the hash key. The node with the smallest returned hash value would then become the regional leader. Since regional members know each other's *id* and location, they can independently execute the hash function to determine which node should be the leader.

Below we briefly describe how the extended region-based group key management protocol handles group partition, group merge, node eviction, group communication, and intrusion detection events. Standard mobile group events such as member join, member leave and mobility induced operations are described in [2] and will not be repeated here.

Group partition: Group members may lose connection with each other due to node failure or node mobility. Thus, a group may be partitioned into multiple groups dynamically. In general, group partition increases as the node density decreases and as the node mobility increases. A group partition event starts with a region being partitioned and detected by members in the region missing the leader's beacon message, and by the leader missing its regional members' beaconing messages. In the former case, a new leader may be elected following a leader election protocol. 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. In any case, all leaders in each partitioned group will execute GDH to agree on a new leader key K_{RL}.

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 non-group members by listening to beacon messages issued from nodes with a different group ID. After authentication a new regional leader in a merged region is selected following the leader selection protocol. Then 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}. Finally, a new group key K_G is generated by all leaders and is distributed to all group members in the merged group.

Eviction: After a node is detected as compromised, the group key K_G is rekeyed based on GDH to evict the compromised node.

Group communication: Each node may communicate with other nodes in the group to request data. We assume that the time interval is exponentially distributed with rate λ_q. A node may disseminate a message to its group members by sending the message to its leader using K_R; the leader would then forward the message to other leaders using K_{RL}; finally, each leader will disseminate the message to all its regional members using K_R based on multicasting.

Intrusion detection: Messages required for IDS activities follow the rules for group communication, including status exchange, vote-participant selection, vote-participant-list dissemination, and vote dissemination. A target node is examined by IDS periodically and if the target node is considered compromised, it will be evicted by rekeying the group key K_G based on GDH.

4. System model

We assume that the GCS is in MANETs in which there is no centralized key server to authenticate and authorize individual group members. Each node is preloaded with the public keys of all other group members for authentication purposes. For efficiency reasons, a group key is used for group communications by group members. The group key is rekeyed by running a *contributory key agreement (CKA)* protocol such as GDH, as there is no centralized trust entity to generate and disseminate the group key. The group members of the proposed GCS in MANETs are assumed to be spread over a geographical area $A = \pi r^2$ with r being the radius.

The workload and operational conditions of a GCS in MANETs can be characterized by a set of model parameters. 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 $\mu/(\lambda + \mu)$. Nodes can move freely with a mobility rate of σ. A group is connectivity-oriented, that is, 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 GCS 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 what its size, acts independently of other groups with the intent to continue with mission execution. Nodes in a group must satisfy the *forward/backward secrecy*, *confidentiality, integrity,* and *authentication* requirements for secure group communications in the presence of outsider and insider attacks. Reliable transmission is a system requirement for secure group communications. We assume that *view synchrony (VS)* is guaranteed in GCSs [2], which guarantees that messages are delivered reliably and in the proper order under the same membership view. That is, a receiver will see the same membership view as viewed by the sender. While maintaining view synchrony introduces many control messages, it will only affect the communication cost incurred, but not *MTTSF* because it won't cause a security failure condition to be satisfied.

We use a hexagonal coverage model [2] to cover the operational area. While there are many ways to divide the operational area into multiple regions, we choose the hexagonal model for analytical solution conveniences. The solution methodology can be applied to other models such as mesh networks without loss of generality. The operational geographical area $A = \pi r^2$ is divided into $R(n) = 3n^2 + 3n + 1$ regions such that there are 61 regions with $n = 4$, 37 regions with $n = 3$, 19 regions with $n = 2$, 7 regions with $n = 1$, and 1 region with $n = 0$, where n is the ring level. The probability that a member moves across a boundary between two regions, denoted by $P_{RM}(n)$, is given by:

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

It can be shown that the regional mobility rate σ_n is given by:

$$\sigma_n = (2n+1)_n P_{RM}(n). \tag{2}$$

For attacker behaviors, we assume the presence of smart inside attackers who will attempt to compromise nodes with a variable rate depending on the number of compromised nodes in the system. Based on the principles given in [41], we assume that the attacker strength increases proportional to the number of compromised nodes in the system because of the collusion of compromised nodes. To dynamically react to the attacker strength, we also adaptively increase the intrusion detection rate proportional to the number of compromised nodes detected to counteract intrusion.

To alleviate collusion, the system performs region-voting-based IDS by which a majority of *m* vote-participants must agree in order to evict a suspicious target node, where *m* is the number of vote-participants randomly selected from group members in the same region of the target node. Whether one should select vote-participants from a larger or smaller region depends on the security and performance requirements of the GCS. The special case in which there is no region is used as a baseline model against which region-voting-based IDS is compared. In general, region-voting-based IDS is characterized by its false negative probability (P_{fn}) and false positive probability (P_{fp}) which depend on *p1* and *p2* in host-based IDS and the number of compromised nodes in the system.

4.1. Security failure conditions

We assume that the GCS enters a security failure state when any mobile group fails. A mobile group fails when one of the two conditions below is true:

Condition C1: a compromised but undetected member requests and subsequently obtains "secret" data using the group key, representing a *loss of integrity* [46] security failure.

Condition C2: more than 1/3 of member nodes in a group are compromised but not detected by IDS, representing a *loss of availability* [46] security failure. This follows the Byzantine Failure model [47] such that when more than 1/3 of member nodes are compromised, the system fails.

The first condition represents that important data are compromised. The second condition represents that the mobile group is unable to function correctly and is compromised as a whole. Both conditions lead to security failure.

4.2. Metrics

We use the following two metrics to measure security and performance properties of a GCS in MANETs:

MTTSF (Mean Time to Security Failure): This metric indicates the lifetime of the GCS before it reaches a security failure state. For a secure GCS, a security failure occurs when either Condition C1 or Condition C2 is *true*. As a security metric, lower *MTTSF* means a faster *loss of system integrity* or *loss of availability*. Therefore, a design goal is to maximize *MTTSF*.

Communication cost($\hat{C}_{total}$): This metric indicates total control message traffic incurred per time unit (s) including group communication, status exchange, rekeying, intrusion detection, beacon, group partition/merge, and mobility-induced activities. A high $\hat{C}_{\mathrm{total}}$ translates into a high level of contention over the wireless channel and consequently a high response time for group communication operations. A design goal is to minimize the average $\hat{C}_{\mathrm{total}}$ by exploring the tradeoff between inter-regional and intra-regional communication costs. The computational cost is not included in $\hat{C}_{\mathrm{total}}$.

4.3. Attack model

We integrate region-based group key management with region-voting-based IDS to deal with both outsider and insider attacks. In general, an *outside attacker* would attempt to gain authorized access and then perpetrate as an inside attacker. Below we discuss possible outsider attack scenarios [48] and our countermeasures. 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. We use individual rekeying [49] at three different levels (i.e., regional key, leader key, and group key) to prevent loss of confidentiality. An outside attacker can also attempt to modify a data packet to break data integrity. We use a symmetric key (the group key) shared by only legitimate group members to prevent loss of data integrity. An outside attacker may impersonate a group member to join a group. We have each node preloaded with public keys of all other group members to ensure source authenticity and to prevent potential impersonation attacks during the authentication process of a new group member's join. Active outsider attacks such as denial of service (DoS) attacks can also be eased by authentication. An outside attacker may forge packets. Since only legitimate group members with a secret key distributed can understand data packets communicated by other group members, forged packets will be discarded. An outsider can also perform jamming attacks. The standard defense against jamming includes spread spectrum or frequency hopping communication, locating the jamming area and rerouting traffic.

Insider attackers are compromised nodes disguising themselves as legitimate healthy members to disrupt the system. We discuss insider attack scenarios and our countermeasures below. An adversary can collude with other compromised nodes so as to more efficiently compromise another node and eventually cause the system to fail. For example, in the process of region-voting-based intrusion detection, an adversary can cast a negative vote against a trusted healthy node or cast a

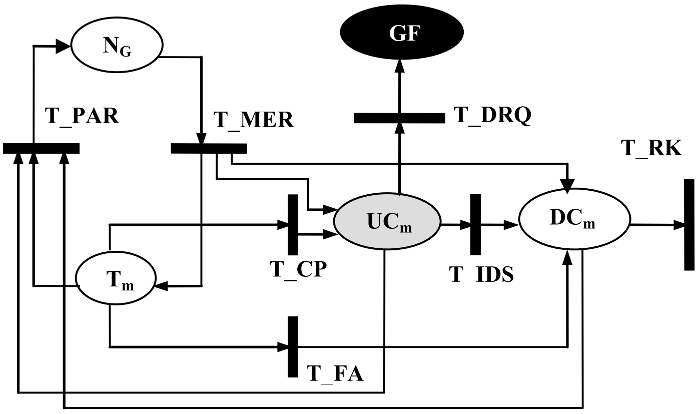


Fig. 2. SPN performance model.

positive vote for a compromised node. An inside attacker aims to obtain secret group information by communicating with other group members with a legitimate group key. Thus, a smart inside attacker may not wish to interrupt the rekeying process so that it can learn of the new group key with which to obtain secret group information. When this happens, a security failure due to Condition C1 occurs. This is countered by invoking IDS at the optimal rate identified by our mathematical model to detect and remove compromised nodes, thus prolonging the system lifetime. A compromised node can share information with other nodes including both outside attackers and inside attackers to more easily compromise other nodes. We model the attacker strength being proportional to the degree of compromised nodes in the system, which can be estimated by our mathematical model. The increase in attacker strength is countered by increasing the IDS detection rate proportional to the number of compromised nodes detected by IDS. When more than 1/3 of the group members have been compromised, a security failure occurs due to Condition C2. This is also countered in our system by invoking IDS to detect and remove compromised nodes.

5. Performance model

We develop a mathematical model based on SPN as shown in Fig. 2 to describe the behaviors of a GCS instrumented with IDS to deal with insider attacks and region-based group key management to deal with outsider attacks in MANETs. The purpose of developing the model is to identify optimal settings to maximize *MTTSF* while satisfying the performance requirement in terms of $\hat{C}_{\text{total}}$. The SPN model is constructed as follows:

The SPN model describes the behavior of a mobile group as it evolves. It suffices to model the behavior of a single mobile group as the GCS fails when a single group fails for mission-critical applications. A mobile group may be partitioned into two and may merge with another group during its lifetime. We track trusted members, compromised members undetected, and compromised members detected during its lifetime to understand its security and performance characteristics.

We use places to classify nodes except for place N_G which holds the current number of groups in the system. Specifically, place T_m holds trusted members, UC_m holds compromised nodes not yet detected by IDS, and DC_m holds compromised nodes that have been detected by IDS. Note that T_m, UC_m, and DC_m represent nodes in one group, not in the system. Thus, the numbers of nodes in places T_m, UC_m, and DC_m, obtained by mark(T_m), mark(UC_m), and mark(DC_m), respectively, would be adjusted based on the number of groups existing in the system (obtained by mark(N_G)), which changes upon group merge/partition events.

We use transitions to model events. The event occurrence times are either deterministic or exponentially distributed in our model. Specifically, *T_FA* models a node being falsely identified as compromised. *T_IDS* models a compromised node being detected. The event occurrence times to *T_FA* and *T_IDS* are fixed periodic times due to the use of the base periodic detection interval T_{IDS}. *T_CP* models a node being compromised. In the reliability community, it is well accepted that the software/hardware failure time follows exponential distribution. Therefore, it is justified to assume that the event occurrence time of *T_CP* is exponentially distributed. *T_MER* and *T_PAR* model the group merge or partition event, respectively. We model group merge and partition events by a *birth–death process* (a Markov model) as shown in Fig. 3 so the time it takes for a group partition or merge event to occur, in *T_MER* or *T_PAR*, is inherently exponentially distributed. We have used deterministic distribution for *T_RK* for modeling the rekeying time in the analysis. *T_DRQ* models a data leak security failure (Condition C1). The assumption that packet arrivals follow a Poisson distribution is well accepted in the networking community and has been used in many previous studies. Therefore, it is justified to assume that the event occurrence time of *T_DRQ* is exponentially distributed.

A firing of a transition will change the state of the system, which is represented by the distribution of tokens in the SPN. For example, mark(N_G) changes upon firing *T_MER* or *T_PAR* since the number of groups is changed upon a group merge or partition event; the number of compromised nodes undetected is incremented by 1 and, thus, place UC_m will hold one more token when *T_CP* fires. A transition is eligible to fire when the firing conditions associated with the event are met. This

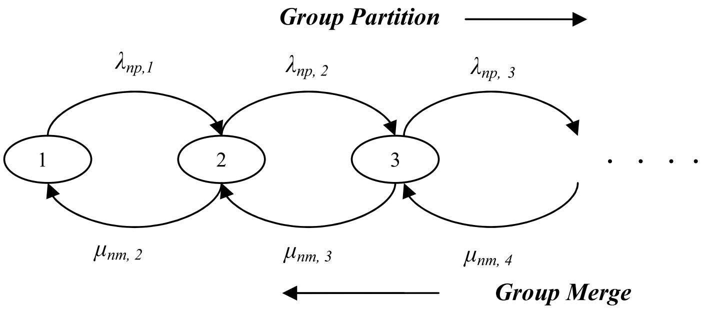


Fig. 3. A birth–death process for group merging/partitioning events.

is presented by (1) its input place must contain at least one token and (2) the associated enabling guard function, if exists, must return *true*. For example, *T_CP* is enabled to fire when there are "good" nodes in the group, that is, place T_m holds at least one token, and the enabling function associated with *T_CP* returns *true*.

Except for tokens contained in place N_G, we use a "token" in the SPN model to represent a node in the group. The population of each type of nodes is equal to the number of tokens in the corresponding place. Initially, all N members are trusted in one group and put in place T_m as tokens.

Trusted members may become compromised because of insider attacks with a node-compromising rate $A(m_c)$. This is modeled by firing transition *T_CP* and moving tokens one at a time (if it exists) from place T_m to place UC_m. Tokens in place UC_m represent compromised but undetected member nodes. We consider the system as having experienced a security failure when data are leaked out to compromised but undetected members, i.e., Condition C1. A compromised but undetected member will attempt to compromise data from other members in the group. This is modeled by associating transition *T_DRQ* with rate $\lambda_q * \text{mark}(UC_m)$. Firing transition *T_DRQ* will move a token into place *GF*, at which point we regard the system as having experienced a security failure due to Condition C1.

A compromised node in place UC_m may be detected by IDS before it compromises data in the GCS. The intrusion detection activity of the system is modeled by the detection function with rate $D(m_d)$. Whether the damage has been done by a compromised node before the compromised node is detected depends on the relative magnitude of the node-compromising rate ($A(m_c)$) versus the IDS detection rate ($D(m_d)$). When transition *T_IDS* fires, a token in place UC_m will be moved to place DC_m, meaning that a compromised, undetected node now becomes detected by IDS. For region-voting-based IDS, the transition rate of *T_IDS* is $\text{mark}(UC_m) * D(m_d) * (1 - P_{fn})$, taking into consideration the false negative probability of region-voting-based IDS used. Region-voting-based IDS can also false-positively identify a trusted member node as compromised. This is modeled by moving a trusted member in place T_m to place DC_m after transition *T_FA* fires with rate $\text{mark}(T_m) * D(m_d) * P_{fp}$. Note that region-voting-based IDS parameters, P_{fn} and P_{fp}, can be derived based on *p1* and *p2*, the number of vote-participants (m), and the current number of compromised nodes which may collude to disrupt the service of the system. Later we will show how we may parameterize P_{fn} and P_{fp}.

Finally, the group is considered as experiencing a security failure if either one of the two security failure conditions, Condition C1 or Condition C2, is met. This is modeled by making the group enter an absorbing state when either Condition C1 or Condition C2 is *true*. In the SPN model, this is achieved by associating every transition in the SPN model with an enabling function that returns *false* (thus disabling the transition from firing) when either Condition C1 or Condition C2 is met, and returns *true* otherwise. For the SPN model, Condition C1 is *true* when $\text{mark}(GF) > 0$ representing that data have been leaked out to compromised, undetected members; Condition C2 is *true* when more than 1/3 of member nodes are compromised but undetected as indicated in Eq. (3) below, where *mark* (UC_m) returns the number of compromised but undetected nodes in the group and $\text{mark}(T_m)$ returns the number of trusted healthy nodes in the group.

$$\frac{\text{mark}(UC_m)}{\text{mark}(T_m) + \text{mark}(UC_m)} > \frac{1}{3}. \tag{3}$$

5.1. Group merge and partition

We model group merge and partition events by a *birth–death process* as shown in Fig. 3. When the system has i groups, i.e., in state i, the group partitioning rate is $\lambda_{np,i}$ and the group merging rate is $\mu_{nm,i}$. We parameterize merging/partitioning rates by means of simulation. We first observe the number of merge and partition events by simulation for a sufficiently long period of time T. We next observe the sojourn time S_i when the system stays in state i, i.e., when i groups are present in the system. Let $N_{nm,i}$ and $N_{np,i}$ be the numbers of group merge and partition events observed in state i, respectively. Then, by first order approximation the merging/partitioning rates in state i, represented by $\mu_{nm,i}$ and $\lambda_{np,i}$, may be estimated by:

$$\mu_{nm,i} = \frac{N_{nm,i}}{S_i} \qquad \lambda_{np,i} = \frac{N_{np,i}}{S_i}. \tag{4}$$

Note that the merging/partitioning rates parameterized in Eq. (4) are a function of the node mobility and density in general. We observe that when node density is high, group merge is more likely to occur than group partition, thus leading to a smaller number of groups observed in the system. On the other hand, as the node density is low, the system is more likely to stay in a state in which there is a large number of groups. In other words, when the node density is low, group partitioning is more likely to occur than group merging.

5.2. Calculation of security and performance metrics

We calculate ***MTTSF*** using the concept of *mean time to absorption* in the SPN model. Specifically, we use a reward assignment such that a reward of 1 is assigned to all states except absorbing states which is modeled based on the two security failure conditions (i.e., if either Condition C1 or Condition C2 is met, the system fails). By this reward assignment, a reward of 1 (time unit) is cumulatively added to the system lifetime with every time unit elapsed until the system fails.

We calculate $\hat{\boldsymbol{C}}_{\boldsymbol{total}}$ by the probability-weighted average of $\hat{C}_{total,i}$ representing the communication cost incurred per time unit (s) in state *i*. Specifically, $\hat{C}_{\text{total}}$ is calculated by accumulating $\hat{C}_{total,i}(t)$ over *MTTSF* divided by *MTTSF*, i.e.,

$$\hat{C}_{\text{total}} = \frac{\int_0^{MTTSF} \hat{C}_{\text{total},i}(t)\mathrm{d}t}{MTTSF}. \tag{5}$$

$\hat{\boldsymbol{C}}_{\boldsymbol{total},\boldsymbol{i}}$ is calculated as:

$$\hat{C}_{\text{total},i} = \hat{C}_{GC,\ i} + \hat{C}_{\text{status},i} + \hat{C}_{\text{rekey},i} + \hat{C}_{IDS,\ i} + \hat{C}_{\text{beacon},i} + \hat{C}_{mp,\ i} + \hat{C}_{\text{mobility},i} \tag{6}$$

where $\hat{C}_{GC,i}$, $\hat{C}_{\text{status},i}$, $\hat{C}_{\text{rekey},i}$, $\hat{C}_{IDS,i}$, $\hat{C}_{beacon,i}$, $\hat{C}_{mp,i}$, and $\hat{C}_{mobility,i}$ are the cost components for group communication, status exchange, rekeying, intrusion detection, beacon, group partition/merge, and mobility events, respectively, given that the number of groups in the system is *i*. The calculation of $\hat{C}_{\text{beacon},i}$, $\hat{C}_{mp,i}$, and $\hat{C}_{\text{mobility},i}$ follows that in [2]. Below we explain how we calculate $\hat{C}_{GC,i}$, $\hat{C}_{\text{status},i}$, $\hat{C}_{\text{rekey},i}$, $\hat{C}_{IDS,i}$ based on the SPN model.

$\hat{\boldsymbol{C}}_{\boldsymbol{GC},\boldsymbol{i}}$: this cost includes the communication cost incurred by group communication activities. It is calculated by:

$$\hat{C}_{GC,i} = \lambda_q \times N \times (b_{GC}/e) \times [N_{\text{region},i} \times H_{\text{region}} \times H_{\text{leader},i}] \tag{7}$$

where λ_q is the group communication rate, *N* is the number of active group members in the single group observed (i.e., $\text{mark}(UC_m) + \text{mark}(T_m)$), b_{GC} is the message size (*bits*) of a group communication packet, *e* is the channel error probability and thus b_{GC}/e is the total number of bits transmitted before the packet is received, $N_{\text{region},i}$ is the number of regions in a group where there are *i* groups, H_{region} is the number of hops multicast from a leader to all regional members, and $H_{\text{leader},i}$ is the number of hops multicast from one leader to all other leaders in a group where there are *i* groups. The number of hops in H_{region} is counted based on the use of a binary tree for a multicasting message. Here A_{region} is the area of a region, s is the circum-radius of a hexagon-shaped region, *R* is the wireless per-hop radio range (*m*), and *i* is the number of groups observed in the system and *R (n)* returns the number of regions in the entire system. When 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. $H_{\text{leader},i}$ returns 0 when $i = 1$ as a special case. H_{region}, $H_{\text{leader},i}$, and $N_{\text{region},i}$ are calculated as follows:

$$H_{\text{region}} = \frac{s}{R} \times (N_{\text{region}}^{\text{members}} - 1) \quad \text{where } s = \sqrt{\frac{2}{3\sqrt{3}} A_{\text{region}}} \quad A_{\text{region}} = \frac{A}{R(n)} \tag{8}$$

$$H_{\text{leader},i} = \frac{r}{R\sqrt{i}} \times (N_{\text{region},i} - 1) \tag{9}$$

$$N_{\text{region},i} = \frac{R(n)}{i}. \tag{10}$$

$\hat{\boldsymbol{C}}_{\boldsymbol{status},\boldsymbol{i}}$: this cost is for group node *status exchange* for intrusion detection. It is calculated by:

$$\hat{C}_{\text{status},i} = \frac{(N \times (b_s/e)) \times [N_{\text{region},i} \times H_{\text{region}} + H_{\text{leader},i}]}{T_{\text{status}}} \tag{11}$$

where T_{status} is the periodic time interval for disseminating a status exchange message, *N* is the number of group members, and b_s is the message size (*bits*) of the status exchange information.

$\hat{\boldsymbol{C}}_{\boldsymbol{rekey},\boldsymbol{i}}$: this cost is for group key rekeying due to join/leave events and forced evictions to evict detected compromised nodes. It is calculated as:

$$\hat{C}_{\text{rekey},i} = \hat{C}_{\text{join/leave},i} + \hat{C}_{\text{eviction},i} \tag{12}$$

where $\hat{C}_{join/leave,i}$ is the cost introduced by leave and join operations per time unit and $\hat{C}_{\text{eviction},i}$ is the cost introduced by forced evictions per time unit. $\hat{C}_{\text{join/leave},i}$ is as calculated in [2] and is not repeated here. $\hat{C}_{\text{eviction},i}$ is calculated by:

$$\hat{C}_{\text{eviction},i} = [rate(T\_IDS) + rate(T\_FA)] \times \hat{C}_{\text{leave},i}. \tag{13}$$

Here *rate(T_IDS)* gives the intrusion detection rate and *rate(T_FA)* gives the false alarm rate which detects trusted nodes as compromised nodes. Both can be obtained easily from the SPN model. $\hat{C}_{\text{leave},i}$ is the cost per leave operation when there are i groups in the system, as calculated in [2].

$\boldsymbol{\hat{C}_{IDS,i}}$: this is the communication cost due to IDS. For region-voting-based IDS, this cost is computed as:

$$\hat{C}_{IDS,\,i} = D(m_d) \times (1 - P_{fn}) \times N \times [b_{m-list} + m \times b_v]/e \times [H^{uni}_{\text{leader},i} + H_{\text{region}} \times N_{\text{region},i}] \tag{14}$$

where $D(m_d)$ is the detection rate proportional to the number of compromised nodes detected by IDS to counter increasing attacker strength, P_{fn} is the probability of false negatives, N is the number of current members in a group, m is the number of vote-participants against a target node, b_{m-list} is the message size (*bits*) of the list containing m vote participants, b_v is the message size (*bits*) of a vote, and $H^{uni}_{\text{leader},i}$ is the number of hops between two leaders calculated as $r/R\sqrt{i}$.

6. Numerical results and analysis

In this section, we present numerical data for *MTTSF* and $\hat{C}_{\text{total}}$ obtained when given a set of parameter values characterizing the operational and environmental conditions and show that there exist optimal design settings in terms of the IDS detection interval and the regional area size under which *MTTSF* is maximized while $\hat{C}_{\text{total}}$ is minimized for a GCS in MANET environments. We first parameterize (i.e., give values to) key model parameters reflecting the operating conditions of the GCS and then we present numerical results and analysis with physical interpretations given.

6.1. Parameterization

Below we describe how to assign proper values to several key model parameters to reflect the operational and environmental conditions of the system.

$\boldsymbol{N}$**:** This is the number of current active group members in the system. This number evolves dynamically as the system evicts compromised nodes. Since a node leaves the group voluntarily with rate μ and joins the group with rate λ, the probability that a node is active in the group is $\lambda/(\lambda + \mu)$ and the probability that it is not is $\mu/(\lambda + \mu)$. Let N_{total} be the total group population at any time. Then, $N = N_{\text{total}}\lambda/(\lambda + \mu)$. In the SPN model, we initially place $(N_{init}/\text{mark}(N_G)) \times (\lambda/(\lambda\ + \mu))$ tokens in place T_m, as the initial population of a mobile group. As the system evolves, N is obtained with $\text{mark}(T_m) + \text{mark}(UC_m)$ indicating the number of current active members in a group.

$\boldsymbol{T_{cm}}$: This is the communication time required for disseminating a rekeying message. The reciprocal of T_{cm} is the rate of transition *T_RK*. As we use GDH for rekeying, T_{cm} can be calculated by $C_{\text{leave},i}$ divided by *BW*, the wireless network bandwidth (*Mbps*) in the MANET.

$\boldsymbol{A(m_c)}$: This is the attacker function that returns the rate at which a node is compromised in the system. It is also the rate of transition *T_CP*. Specifically, $A(m_c) = \lambda_c \times m_c$ where λ_c is the base compromising rate and m_c represents the *degree* of compromised nodes currently in the system, defined by the ratio of N to the number of good nodes, i.e., $m_c = (\text{mark}(UC_m) + \text{mark}(T_m))/\text{mark}(T_m)$. The base compromising rate (λ_c) can be obtained by first-order approximation from observing the number of compromised nodes over a time period.

$\boldsymbol{D(m_d)}$: This is the detection function [41] that returns the rate at which intrusion detection is invoked, adjusted based on the accumulated number of nodes that have been detected by IDS. It is also the transition rate of *T_IDS* in our SPN model. Specifically, $D(m_d) = m_d/T_{IDS}$, where T_{IDS} is the base intrusion detection interval and m_d represents the *degree* of nodes that have been detected by IDS, defined by the ratio of N_{init} to N, i.e., $m_d = N_{init}/(\text{mark}(UC_m) + \text{mark}(T_m))$. The base intrusion detection interval T_{IDS} is a design parameter to be adjusted to maximize *MTTSF* while minimizing $\hat{C}_{\text{total}}$.

$\boldsymbol{P_{fn}}$ **&** $\boldsymbol{P_{fp}}$: P_{fn} is the probability of false negatives and P_{fp} is the probability of false positives based on vote-based IDS. We parameterize them by:

$$P_{fp} \text{ or } P_{fn} = \sum_{i=0}^{m-N_{\text{majority}}} \left[\frac{C\begin{pmatrix} N_{\text{bad}} \\ N_{\text{majority}} + i \end{pmatrix} \times C\begin{pmatrix} N_{\text{good}} \\ m - (N_{\text{majority}} + i) \end{pmatrix}}{C\begin{pmatrix} N_{\text{good}} + N_{\text{bad}} \\ m \end{pmatrix}} \right]$$
$$+ \sum_{i=0}^{m-N_{\text{majority}}} \left[\frac{C\begin{pmatrix} N_{\text{bad}} \\ i \end{pmatrix} \times \sum\limits_{j=N_{\text{majority}}-i}^{m-i} \left[C\begin{pmatrix} N_{\text{good}} \\ j \end{pmatrix} \times p^j \times C\begin{pmatrix} N_{\text{good}} - j \\ m-i-j \end{pmatrix} \times (1-p)^{(m-i-j)} \right]}{C\begin{pmatrix} N_{\text{good}} + N_{\text{bad}} \\ m \end{pmatrix}} \right]. \tag{15}$$

Here p corresponds to *p1* or *p2* for false negative or false positive probability for P_{fn} or P_{fp}. In region-voting-based IDS, $N_{\text{good}} = \text{mark}(T_m)/N_{\text{region},i}$ and $N_{\text{bad}} = \text{mark}(UC_m)/N_{\text{region},i}$. Basically, P_{fn} is calculated by the number of compromised nodes incorrectly diagnosed as trusted healthy nodes (i.e., detecting a bad node as a good node) over the number of detected nodes. On the other hand, P_{fp} is calculated by the number of normal nodes incorrectly flagged as anomalies over the number

Table 2
Main parameters and default values.

Parameter	Value	Parameter	Value	Parameter	Value
λ	1/(60*60)	σ	1/(60*60*32)	T_{RB}	5 s
μ	1/(60*60*4)	BW	1 Mbps	T_{LB}	2 s
T_{IDS}	5–1200 s	N_{init}	150 nodes	m	3
T_{status}	300 s	$D(m_d)/A(m_c)$	Linear	R	200 m
λ_c	1/(60*60*24)	r	500 m	b_{vote}	100 bits
λ_q	1/30	b_{GDH}	64 bits	b_{m-list}	100 bits
$p1$	2%	b_{GC}	800 bits	e	0.5
$p2$	2%	b_s	400 bits		

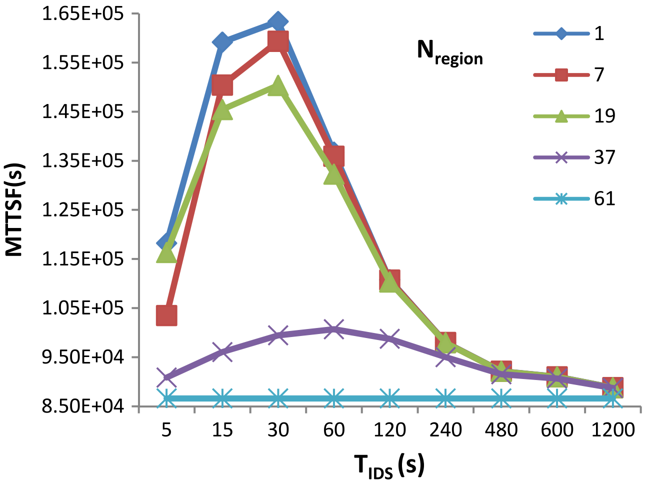


Fig. 4. *MTTSF* versus T_{IDS} with $p1 = p2 = 0.005$.

of detected normal nodes. We consider intrinsic defect of host-based IDS in each node as well as collusion of compromised nodes in region-voting-based IDS, so a compromised participant can cast a negative vote against a healthy target node and can cast a positive vote for a malicious node.

6.2. Data and analysis

We vary values of key design parameters to analyze their effects on optimal settings (in terms of the best regional size and the best IDS interval) under which the system performance is optimized. Table 2 summarizes default parameter values. We test the effects of key parameters such as *per-node* false negative or false positive probabilities ($p1$, $p2$), the group communication rate (λ_q) and the compromising rate (λ_c). Note that for region-voting-based IDS, P_{fn} and P_{fp} are calculated based on Eq. (15).

Figs. 4 and 5 analyze the optimal settings in terms of the IDS detection interval T_{IDS} and the number of regions N_{region} under which *MTTSF* is maximized. Fig. 4 is for the case of low $p1$ and $p2$ values while Fig. 5 is for the case of high $p1$ and $p2$ values. The special case in which there is only one region ($N_{\text{region}} = 1$) is also considered.

From Figs. 4 and 5, we observe that there exists an optimal T_{IDS} that maximizes *MTTSF*. In general, as T_{IDS} increases, *MTTSF* increases until its optimal T_{IDS} is reached, and then *MTTSF* decreases past the optimal T_{IDS}. The reason of decreasing *MTTSF* after reaching the optimal point is that the false positive probability (P_{fp}) increases as T_{IDS} decreases, therefore resulting in more nodes being falsely identified as compromised and being evicted from the system. Here we note that decreasing T_{IDS} (thus performing IDS more often) in effect increases the number of good nodes being misdiagnosed as bad nodes over time, thereby decreasing the number of good nodes (N_{good}) and adversely affecting the false positive probability computed by Eq. (15). Next, we observe that there exists an optimal regional area size which is largely dictated by $p1$ and $p2$ values. Note that P_{fp} is one aspect of false alarms generated by IDS, which increases if IDS is more frequently triggered. When $p1 = p2$ is sufficiently low (i.e., $p1 = p2 = 0.005$) as shown in Fig. 4, the best *MTTSF* is found with $N_{\text{region}} = 1$ at $T_{IDS} = 30$ s. When $p1 = p2$ is sufficiently high (i.e., $p1 = p2 = 0.02$) as shown in Fig. 5, the best *MTTSF* is identified with $N_{\text{region}} = 19$ at $T_{IDS} = 120$ s. The optimal *MTTSF* exists due to the tradeoff between the positive and adverse effects of performing IDS.

Specifically, when $p1 = p2$ is sufficiently low, P_{fp} and P_{fn} are also sufficiently low due to the positive effect of high quality IDS, so the system benefits the best by using m vote-participants out of a large region, i.e., at $N_{\text{region}} = 1$. On the other hand, when $p1 = p2$ is high, P_{fp} and P_{fn} are also sufficiently high due to the adverse effect of IDS, so the system benefits the best by using m vote-participants out of a moderately large region at $N_{\text{region}} = 19$ such that it still has a good chance of finding m vote-participants in the region without suffering too much from high P_{fp} and P_{fn}.

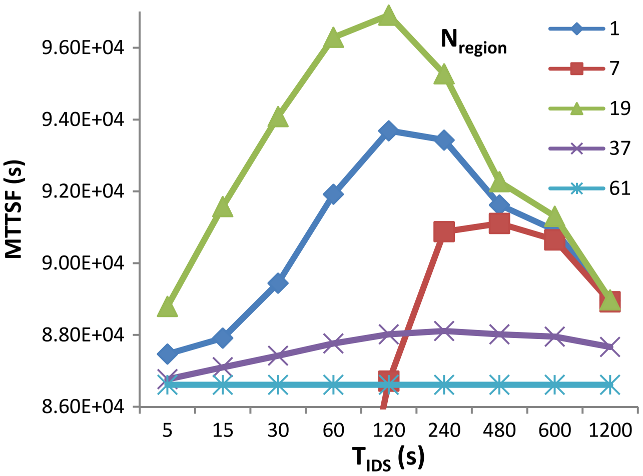
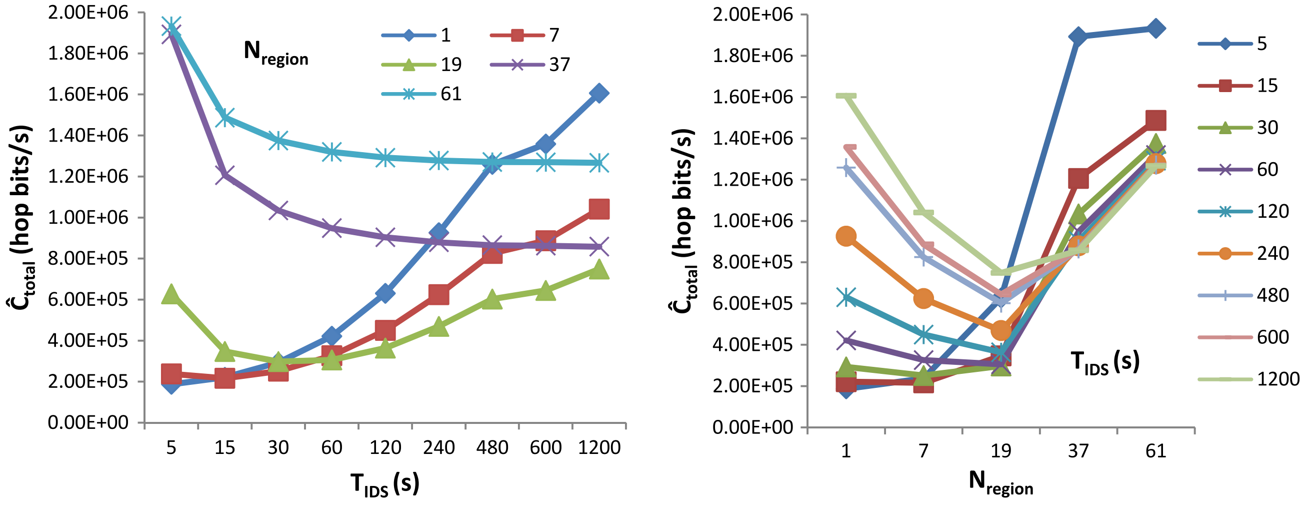


Fig. 5. *MTTSF* versus T_{IDS} with $p1 = p2 = 0.02$.



(a) $\hat{C}_{\text{total}}$ versus T_{IDS} with $p1 = p2 = 0.02$.

(b) $\hat{C}_{\text{total}}$ versus N_{region} with $p1 = p2 = 0.02$.

Fig. 6. $\hat{C}_{\text{total}}$ with respect to T_{IDS} and N_{region} with $p1 = p2 = 0.02$.

Lastly, we notice that as $p1 = p2$ increases, the optimal T_{IDS} increases. This is again due to the tradeoff between the positive and adverse effects of IDS. When high $p1 = p2$ is used, triggering IDS less frequently will generate less false alarms. On the other hand, when sufficiently low $p1 = p2$ is used, triggering of IDS often will improve *MTTSF* with the high quality IDS without generating frequent false alarms. We notice that at $N_{\text{region}} = 61$ *MTTSF* is low and flat because of the very low probability of finding m vote-participants in a small region, resulting in IDS not being used much in the system.

Next we analyze the optimal settings in terms of T_{IDS} and N_{region} under which $\hat{C}_{\text{total}}$ is minimized. Fig. 6(a) shows $\hat{C}_{\text{total}}$ versus T_{IDS} with varying N_{region} while Fig. 6(b) shows $\hat{C}_{\text{total}}$ versus N_{region} with varying T_{IDS}. We observe that $\hat{C}_{\text{total}}$ varies depending on N_{region} and that there is an optimal T_{IDS} under which $\hat{C}_{\text{total}}$ is maximized. Moreover, when T_{IDS} is sufficiently large, say $T_{IDS} > 30$ s, the optimal N_{region} is 19. When $T_{IDS} = 15$ s or 30 s, the optimal N_{region} is at 7. The reason is that when T_{IDS} is sufficiently small, the node density tends to decrease rapidly because of frequent intrusion detection activities to evict compromised nodes. In this case, the system tends to favor a small number of regions (represented by $N_{\text{region}} = 7$) to reduce the inter-regional overhead, as in this case the inter-regional overhead will dominate the intra-regional overhead since the intra-regional overhead will be relatively small when the node density is low. Lastly when $N_{\text{region}} > 37$, $\hat{C}_{\text{total}}$ increases again because the inter-regional communications cost outweighs the intra-regional communication cost. We also observe that in Fig. 6(a) region-voting-based IDS with no region (i.e., $N_{\text{region}} = 1$) at its optimal settings ($T_{IDS} = 5$, $N_{\text{region}} = 1$) performs the best due to a significant reduction of active members because of high false positives generated by IDS.

Next we analyze the effect of $p1$ and $p2$ on the optimal settings that maximize *MTTSF* and minimize $\hat{C}_{\text{total}}$. Figs. 7 and 8 summarize the results. Here for each *MTTSF* or $\hat{C}_{\text{total}}$ curve, T_{IDS} is chosen at its optimal value to isolate out of its effect. As shown in Fig. 7, when $p1 = p2$ is sufficiently low (0.005 or 0.007 or 0.01), *MTTSF* is the best at $N_{\text{region}} = 1$. However, when $p1 = p2$ becomes higher, say $p1 = p2 > 0.01$, we observe the best *MTTSF* at $N_{\text{region}} = 19$. The results correlate well with the results presented earlier in Figs. 4 and 5, so the same physical interpretation applies for the tradeoff between positive

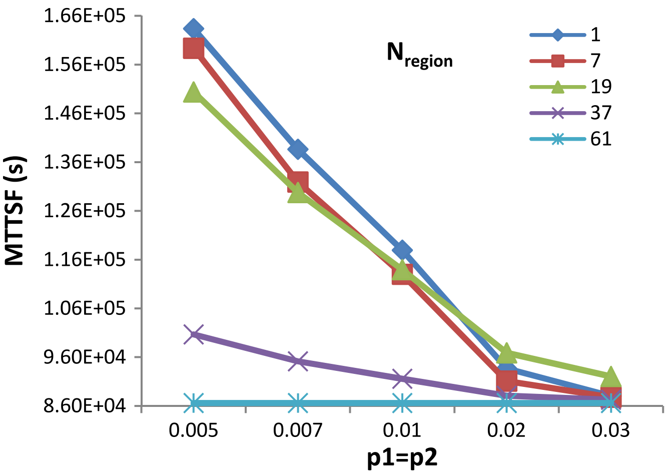


Fig. 7. MTTSF versus *p1* and *p2* with varying N_{region}.

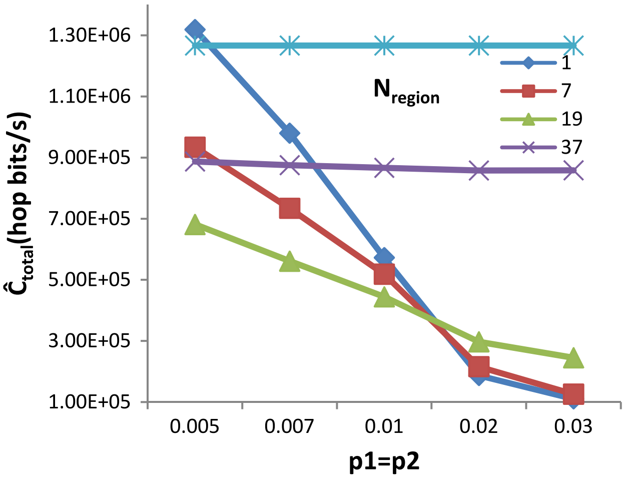


Fig. 8. $\hat{C}_{\text{total}}$ versus *p1* and *p2* with varying N_{region}.

and negative effects of IDS. In Fig. 8, we observe that when $p1 = p2$ is sufficiently low, $\hat{C}_{\text{total}}$ is minimized at $N_{\text{region}} = 19$. However, when $p1 = p2$ becomes higher, say $p1 = p2 > 0.01$, we observe that $\hat{C}_{\text{total}}$ is minimized at $N_{\text{region}} = 7$ or 1. Again this result correlates well with those presented in Fig. 6. We note that there exists a tradeoff between security versus performance. While *MTTSF* is maximized at $N_{\text{region}} = 19$ when $p1 = p2$ is high (e.g., >0.01), $\hat{C}_{\text{total}}$ is minimized at $N_{\text{region}} = 19$ only when $p1 = p2$ is low (e.g., ≤ 0.01). Consequently, the system designer should select the best settings under which the system requirements on *MTTSF* and $\hat{C}_{\text{total}}$ are best satisfied.

Here we note that a major advantage gained from combining IDS with hierarchical key management is that intrusion detection can directly leverage the region-based key management infrastructure to efficiently implement IDS functionality with the region-voting-based IDS design. The case in which intrusion detection and key management are separately managed can be represented by the special case in which there is no concept of region-based management, i.e., $N_{\text{region}} = 1$ with the flat architecture. We see from Figs. 6–8 that our integrated region-based scheme at identifying optimal settings can significantly outperform the no-region special case in the total network traffic incurred per time unit. For example, in Fig. 6(a) when $T_{IDS} > 60$ s, we see that the optimal setting is at $N_{\text{region}} = 19$ for minimizing the overall traffic per time unit under which our integrated region-based scheme significantly performs better the special case in which separate intrusion detection and key management are being employed with $N_{\text{region}} = 1$.

Finally, we analyze the effect of the group communication rate (λ_q) and compromising rate (λ_c) on *MTTSF* or $\hat{C}_{\text{total}}$. Table 3 summarizes the optimal N_{region} value under which *MTTSF* is maximized and/or $\hat{C}_{\text{total}}$ is minimized. We fix all other parameters at their default values. We observe that as λ_q increases, the optimal N_{region} value increases, while the resulting *MTTSF* decreases. The reason for a low *MTTSF* when λ_q is high is mostly due to security failure in Condition C1. We observe that as λ_q increases, $\hat{C}_{\text{total}}$ increases because of the increased group communication cost.

For the effect of λ_c, we observe that as λ_c increases, the optimal N_{region} value decreases while the resulting *MTTSF* decreases. The reason is that with a high λ_c there are more compromised nodes in the system and the system will benefit more from using a smaller N_{region} to increase the probability of being able to find *m* vote-participants in order to more

Table 3
Effect of λ_q and λ_c on optimal N_{region}, *MTTSF* and $\hat{C}_{\text{total}}$.

λ_q	$(N_{\text{region}}, MTTSF)$	$(N_{\text{region}}, \hat{C}_{\text{total}})$	λ_c	$(N_{\text{region}}, MTTSF)$	$(N_{\text{region}}, \hat{C}_{\text{total}})$
Once per 15 s	**(19, 92459)**	(1, 216431)	Once an hr	(1, 9618)	**(19, 855480)**
Once per 30 s	**(19, 96909)**	(1, 186933)	Once per 12 h	(1, 54515)	**(7, 326509)**
Once per 1 min	**(19, 103919)**	**(1, 172210)**	Once a day	**(19, 96909)**	(1, 186933)
Once per 5 min	(1, 145390)	**(19, 83969)**	Once per 2 days	**(19, 184938)**	(1, 106724)
Once per 10 min	(1, 177034)	**(19, 64544)**	Once per 4 days	**(19, 359233)**	(1, 64329)

Table 4
Optimal settings of region-based group key management integrated with voting-based IDS.

		Optimal settings for maximizing *MTTSF*	Optimal settings for minimizing $\hat{C}_{\text{total}}$
Host-based IDS false negative/false positive probabilities (*p1*/*p2*)	High	Small regional area size Large T_{IDS}	Large regional area size Large T_{IDS}
	Low	Large regional area size Small T_{IDS}	Small regional area size Small T_{IDS}
Group communication rate (λ_q)	High	Small regional area size Small T_{IDS}	Large regional area size Small T_{IDS}
	Low	Large regional area size Large T_{IDS}	Small regional area size Large T_{IDS}
Compromising rate (λ_c)	High	Large regional area size Small T_{IDS}	Small regional area size Small T_{IDS}
	Low	Small regional area size Large T_{IDS}	Large regional area size Large T_{IDS}

effectively evict compromised nodes. We also observe that when λ_c increases, the optimal N_{region} in minimizing $\hat{C}_{\text{total}}$ increases while the resulting $\hat{C}_{\text{total}}$ increases slightly. The reason that $\hat{C}_{\text{total}}$ increases slightly as λ_c increases is that when there are more compromised nodes in a group, IDS is triggered more frequently, thus increasing the overall cost. The reason that the optimal N_{region} value increases when λ_c increases is that the inter-regional communication overhead in $\hat{C}_{IDS,i}$ (disseminating a message from a member to a leader) is small compared with the intra-regional communication overhead (disseminating a message from each leader to its regional members), so the system favors a large N_{region} to reduce the intra-regional overhead. Here again we see that a tradeoff exists between security versus performance. The system designer should select the best N_{region} such that both the security requirement (in terms of *MTTSF*) and performance requirement (in terms of $\hat{C}_{\text{total}}$) can be best satisfied.

Table 4 summarizes optimal settings of our proposed integrated hierarchical group key management protocol integrated with region-voting-based IDS, under which *MTTSF* is maximized and/or $\hat{C}_{\text{total}}$ is minimized. This provides guidelines for system designers to fine-tune the regional area size and the IDS detection interval to best satisfy application-imposed security and performance requirements. The general trend is that when (*p1, p2*), λ_q or λ_c is low, low *MTTSF* and high $\hat{C}_{\text{total}}$ are observed, and vice versa. For *MTTSF*, the optimal regional area size and the optimal intrusion detection interval identified are dictated by the tradeoff between positive effects of IDS (e.g., removing compromised nodes as soon as possible not to be vulnerable to them) and negative effects of IDS (e.g., false positives and negatives generated by triggering IDS). For $\hat{C}_{\text{total}}$, the optimal regional area size is primarily determined by the tradeoff between inter-regional and intra-regional communication overheads while the optimal intrusion detection interval is dictated by the tradeoff between IDS related communication cost and group communication cost.

6.3. Comparative analysis

In this section, we perform a comparative analysis of regional-based integrated management against two baseline schemes: (a) a separate management scheme in which intrusion detection and group key management are being employed separately, with $m = 3$ nodes being selected for performing voting-based IDS functions, and (b) a separate management scheme in which intrusion detection and key management are being employed separately, with only one node being selected for IDS functions, i.e., without voting-based IDS.

Figs. 9 and 10 compare *MTTSF* and $\hat{C}_{\text{total}}$ obtained for these three schemes as a function of the host-based IDS false negative/false positive probabilities ($p1/p2$), with T_{IDS} fixed at its optimal value under which the *MTTSF* of the integrated management scheme is maximized. As shown in Fig. 9, region-based integrated management has the best *MTTSF* among all. Furthermore, the difference in *MTTSF* is more significant as the host-based IDS false probability increases. On the other hand, separate management without voting-based IDS performs significantly worse than others due to the fact that it relies on only a single evaluator to make diagnosis decisions. From Fig. 10, we observe that the *MTTSF* advantage gained by integrated

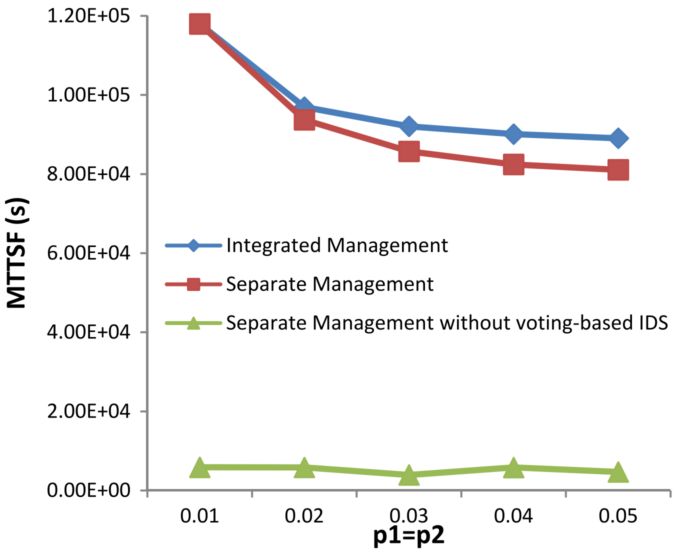


Fig. 9. *MTTSF* of integrated management versus separate management schemes.

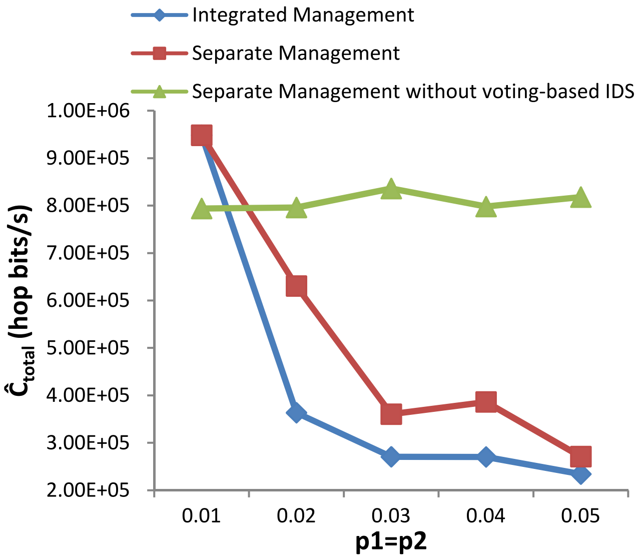


Fig. 10. $\hat{C}_{\text{total}}$ of integrated management versus separate management schemes.

management over separate management does not come with the penalty of increased control message traffic, because integrated management can directly leverage the region-based hierarchical infrastructure to efficiently implement both key management and IDS functions. Together with Figs. 9 and 10, we conclude that integrated management yields higher *MTTSF* while producing lower control message traffic compared with separate deployment of key management and IDS techniques.

7. Conclusion

In this paper, we proposed and analyzed region-based hierarchical group key management integrated with voting-based IDS to deal with both outsider and insider security attacks for a GCS in MANETs for efficiency, scalability and survivability. Our results showed that there exist optimal settings in terms of the optimal regional area size and IDS intrusion detection interval to maximize the *mean time to security failure* while minimizing the *total communication cost* of the GCS. Furthermore, we showed that our proposed region-based integrated scheme outperforms existing schemes for which intrusion detection and group key management are being employed separately. The tradeoff between security and performance can be summarized by a system designer into a lookup table listing optimal settings. Then at runtime for a given set of parameter values characterizing the operational and environmental conditions of MANETs observed dynamically, the GCS can perform a table lookup operation to select optimal settings for maximizing the *mean time to security failure* (representing the lifetime of the GCS) while minimizing the *total communication cost*.

In the future, we plan to investigate if trust-based IDS techniques can be applied to better cope with collusion of compromised nodes. Also in this paper, we considered the use of PKI with preloaded public keys of all group members for source authenticity in the bootstrapping period. As future work, we plan to investigate the use of dynamic public key management and trust-based management for source authenticity.

References

[1] A. Mishra, K. Nadkarni, A. Patcha, Intrusion detection in wireless ad-hoc networks, IEEE Wireless Communications 11 (1) (2004) 48–60.

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

[3] J.H. Cho, I.R. Chen, Performance analysis of distributed intrusion detection protocols for mobile group communication systems, in: IEEE Int'l Symposium on Parallel and Distributed Processing-Workshop, Rome, Italy, May 2009, pp. 1–8.

[4] J.H. Cho, I.R. Chen, P.G. Feng, Effect of intrusion detection on reliability of mission-oriented mobile group systems in mobile ad-hoc networks, IEEE Transactions on Reliability 59 (1) (2010) 231–241.

[5] G. Ciardo, R.M. Fricks, J.K. Muppala, K.S. Trivedi, SPNP users manual version 6, in: Department Electrical Engineering, Duke University, 1999.

[6] 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.

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

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

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

[10] S. Mittra, Iolus: a framework for scalable secure multicasting, in: ACM SIGCOMM, vol. 27, no. 4, Cannes, France, Oct. 1997, pp. 277–288.

[11] C. Duma, N. Shahmehri, P. Lambrix, A hybrid key tree scheme for multicast to balance security and efficiency requirements, in: 12th Int'l Workshop Enabling Technologies: Infrastructure for Collaborative Enterprises, June 2003, pp. 208–213.

[12] A. Balasubramanian, S. Mishra, R. Sridhar, Analysis of a hybrid key management solution for ad-hoc networks, in: IEEE Wireless Communications and Networking Conf., vol. 4, 2005, pp. 2082–2087.

[13] S. Banerjee, B. Bhattacharjee, Scalable secure group communication over IP multicast, IEEE Journal on Selected Areas in Communications 20 (8) (2002) 1511–1527.

[14] S. Basagni, Distributed clustering for ad-hoc networks, in: Int'l Symposium on Parallel Architectures, Algorithms and Networks, IEEE Computer Society, Australia, June 1999, pp. 310–315.

[15] M. Bechler, H.J. Hof, D. Kraft, F. Pahlke, L. Wolf, A cluster-based security architecture for ad-hoc networks, in: 23rd IEEE INFOCOM, vol. 4, March 2004, pp. 2393–2403.

[16] L. Lazos, R. Poovendran, Energy-aware secure multicast communication in Ad-hoc networks using geographic location information, in: IEEE Int'l Conf. on Acoustics Speech and Signal Processing, vol. 4, April 2003, pp. 201– 204.

[17] J.H. Li, R. Levy, M. Yu, B. Bhattacharjee, A scalable key management and clustering scheme for ad-hoc networks, in: 1st ACM Int'l Conf. on Scalable Information Systems, vol. 152, No. 28, Hong Kong, May 2006, pp. 1–10.

[18] K.H. Rhee, Y.H. Park, T. Gene, An architecture for key management in hierarchical mobile Ad-hoc networks, Journal of Communications and Networks 6 (2) (2004) 156–162.

[19] Y. Wang, X. Li, O. Frieder, Efficient hybrid key agreement protocol for wireless ad-hoc networks, in: 11th Int'l Conf. on Computer Communications and Networks, Miami, FL, Oct. 2002, pp. 404–409.

[20] Y.R. Yang, X. Li, X. Zhang, S.S. Lam, Reliable group rekeying: a performance analysis, in: ACM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication, San Diego, CA, Aug. 2001, pp. 27–38.

[21] B. Sun, K. Wu, U.W. Pooch, Alert aggregation in mobile ad-hoc networks, in: ACM Workshop on Wireless Security, San Diego, CA, Sept. 2003, pp. 69–78.

[22] B. Sun, K. Wu, U.W. Pooch, Routing anomaly detection in mobile Ad-hoc networks, in: 12th IEEE Int'l Conf. on Computer Communications and Networks, Oct. 2003, p. 25–31.

[23] S. Marti, T. Giuli, K. Lai, M. Baker, Mitigating routing misbehavior in mobile ad-hoc networks, in: 6th Annual ACM/IEEE Mobile Computing and Networking, Boston, Massachusetts, Aug. 2000, pp. 255–265.

[24] H. Debar, A. Wespi, Aggregation and correlation of intrusion-detection alerts, in: 4th Int'l Symposium Recent Advances in Intrusion Detection, Davis, CA, Oct. 2001, pp. 85–103.

[25] J.B.D. Cabrera, C. Gutierrez, R.K. Mehra, Infrastructures and algorithms for distributed anomaly-based intrusion detection in mobile ad-hoc networks, in: IEEE Military Communications Conf., vol. 3, Oct. 2005, pp. 1831–1837.

[26] Y. Huang, W. Lee, A cooperative intrusion detection system for ad-hoc networks, in: 1st ACM Workshop on Security of Ad-hoc and Sensor Networks, Fairfax, VA, Oct. 2003, 135–147.

[27] O. Kachirski, R. Guha, Intrusion detection using mobile agents in wireless ad-hoc networks, in: IEEE Workshop on Knowledge Media Networking, July 2002, pp. 153–158.

[28] D. Sterne, P. Balasubramanyam, D. Carman, B. Wilson, R. Talpade, C. Ko, R. Balupari, C.Y. Tseng, T. Bowen, A general cooperative intrusion detection architecture for MANETs,, in: 3rd IEEE Int'l Workshop on Information Assurance, College Park, MD, March 2005, pp. 57–70.

[29] D. Subhadrabandhu, S. Sarkar, F. Anjum, Efficacy of misuse detection in Ad-hoc networks, in: 1st Annual IEEE Communications Society Conf. on Sensor and ad-hoc Communications and Networks, Oct. 2004, pp. 97–107.

[30] D. Subhadrabandhu, S. Sarkar, F. Anjum, A framework for misuse detection in ad-hoc networks-Part I, IEEE Journal on Selected Areas in Communications 24 (2) (2006) 274–289.

[31] D. Subhadrabandhu, S. Sarkar, F. Anjum, A framework for misuse detection in ad-hoc networks-Part II, IEEE Journal on Selected Areas in Communications 24 (2) (2006) 290–304.

[32] D.M. Nicol, W.H. Sanders, K.S. Trivedi, Model-based evaluation: from dependability to security, IEEE Transactions on Dependability and Secure Computing 1 (1) (2004) 48–65.

[33] M. Dacier, Y. Deswarte, M. Kaâniche, Quantitative assessment of operational security: models and tools, Technical Report 96493, Laboratory for Analysis and Architecture of Systems, May 1996.

[34] E. Jonsson, T. Olovsson, A quantitative model of the security intrusion process based on attacker behavior, IEEE Transactions on Software Engineering 23 (4) (1997) 235–245.

[35] K.G. Popstojanova, F. Wang, R. Wang, F. Gong, K. Vaidyanathan, K.S. Trivedi, B. Muthusamy, Characterizing intrusion tolerant systems using a state transition model, in: DARPA Information Survivability Conf. and Exposition, vol. 2, June 2001, pp. 211– 221.

[36] B. Madan, K.G. Popstojanova, K. Vaidyanathan, K.S. Trivedi, A method for modeling and quantifying the security attributes of intrusion tolerant systems, Performance Evaluation 56 (1–4) (2004) 167–186.

[37] B. Madan, K.G. Popstojanova, K. Vaidyanathan, K.S. Trivedi, Modeling and quantification of security attributes of software systems, in: Int'l Conf. Dependable Systems and Networks, Washington DC, June 2002, pp. 505–514.

[38] F. Stevens, T. Courtney, S. Singh, A. Agbaria, J.F. Meyer, W.H. Sanders, P. Pal, Model-based validation of an intrusion-tolerant information system, in: 23rd IEEE Symposium Reliable Distributed Systems, Nübcrnberg, Germany, 18–20 Oct. 2004, pp. 184–194.

[39] D. Wang, D.W. Bharat, B. Madan, K.S. Trivedi, Security analysis of SITAR intrusion tolerance system, in: ACM Workshop on Survivable and Self-regenerative Systems, Fairfax, VA, Oct. 2003, pp. 23–32.
[40] D.J. Leversage, E. James, Estimating a system's mean time-to-compromise, IEEE Security and Privacy 6 (1) (2008) 52–60.
[41] H. Chan, V.D. Gligor, A. Perrig, G. Muralidharan, On the distribution and revocation of cryptographic keys in sensor networks, IEEE Transactions on Dependable and Secure Computing 2 (3) (2005) 233–247.
[42] Y. Zhang, W. Lee, Y.A. Huang, Intrusion detection techniques for mobile wireless networks, Wireless Networks 9 (5) (2003) 545–556.
[43] 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.
[44] 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.
[45] M. Steiner, G. Tsudik, M. Waidner, Diffie–Hellman key distribution extended to group communication, in: 3rd ACM Conf. on Computer and Communications Security, Jan. 1996, pp. 31–37.
[46] T. Karygiannis, L. Owens, Wireless network security: 802.11, bluetooth and handheld devices, National Institute of Standards and Technology (NIST) special publication, 2002, pp. 800–848.
[47] F.C. Gärtner, Byzantine failures and security: arbitrary is not (always) random, Technical Report IC/2003/20, EPFL, April 2003.
[48] K. Inkinen, New secure routing in ad-hoc networks: study and evaluation of proposed schemes, in: Seminar on Internetworking, Sjökulla, Finland, Spring 2004.
[49] X. Li, Y.R. Yang, M.G. Gouda, S.S. Lam, Batch rekeying for secure group communications, in: 10th Int'l Conf. on World Wide Web, Hong Kong, May 2001, pp. 525–534.

Further reading

[1] P. Brutch, C. Ko, Challenges in intrusion detection for wireless ad-hoc networks, in: Symposium on Applications and the Internet Workshops, Jan. 2003, pp. 178–373.

Jin-Hee Cho received the BA from the Ewha Womans University, Seoul, Korea in 1997 and the MS and Ph.D. degrees in Computer Science from Virginia Tech in 2004 and 2008 respectively. She is currently a Computer Scientist at the US Army Research Laboratory (US ARL), Adelphi Research Center, Maryland. Her research interests include wireless mobile networks, mobile ad-hoc networks, sensor networks, secure group communications, group key management, network security, intrusion detection, performance analysis, trust management, cognitive networks, and social networks.

Ing-Ray Chen received the BS degree from the National Taiwan University, Taipei, Taiwan, and the MS 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, wireless systems, dependable and secure computing, multimedia, sensor networks, data and service management, trust management, and reliability and performance analysis. Dr. Chen currently serves as an editor for *Wireless Personal Communications*, *Wireless Communications and Mobile Computing*, *The Computer Journal*, *Security and Network Communications*, and *International Journal on Artificial Intelligence Tools*. He is a member of the IEEE and ACM.