Wireless Pers Commun (2011) 60:725–750
DOI 10.1007/s11277-010-9971-1

Model-Based Evaluation of Distributed Intrusion Detection Protocols for Mobile Group Communication Systems

Jin-Hee Cho · Ing-Ray Chen

Published online: 18 April 2010

Abstract Under highly security vulnerable, resource-restricted, and dynamically changing mobile ad hoc environments, it is critical to be able to maximize the system lifetime while bounding the communication response time for mission-oriented mobile groups. In this paper, we analyze the tradeoff of security versus performance for distributed intrusion detection protocols employed in mobile group communication systems (GCSs). We investigate a distributed voting-based intrusion detection protocol for GCSs in multi-hop mobile ad hoc networks and examine the effect of intrusion detection on system survivability measured by the mean time to security failure (MTTSF) metric and efficiency measured by the communication cost metric. We identify optimal design settings under which the MTTSF metric can be best traded off for the communication cost metric or vice versa. We conduct extensive simulation to validate analytical results obtained. This work provides a general model-based evaluation framework for developing and analyzing intrusion detection protocols that can dynamically adapt to changing attacker strengths with the goal of system lifetime optimization and/or communication cost minimization.

Keywords Model-based evaluation · Intrusion detection · Key management · Group communication systems · Mean time to security failure · False positives · False negatives · Mobile ad hoc networks

J.-H. Cho (✉)
Computational and Information Sciences Directorate (CISD), US Army Research Laboratory (USARL), 2800 Powder Mill Rd., Adelphi, MD 20783, USA
e-mail: jinhee.cho@us.army.mil

I.-R. Chen
Department of Computer Science, Virginia Polytechnic Institute and State University, 7054 Haycock Road, Falls Church, VA 22043, USA
e-mail: irchen@vt.edu

1 Introduction

Developing network security protocols in mobile ad hoc networks (MANETs) has design challenges due to high security vulnerabilities and unique characteristics of MANET environments such as open medium, dynamic changing network topology, decentralized decision-making and cooperation, lack of centralized authority, lack of resources in mobile devices (e.g., bandwidth, memory, computational power), and no clear line of defense [2,27]. Three types of actions against attacks can be taken, namely, prevention, detection, and recovery. Intrusion prevention techniques (e.g., encryption or authentication) can be employed in MANETs to reduce intrusion. However, security holes cannot be perfectly eliminated. Due to the above reason, intrusion detection mechanisms are used as a second line of defense and have become essential for systems with the goal of high-survivability.

This paper concerns distributed intrusion detection employed in mission-oriented group communication systems (GCSs) of MANETs for detecting and evicting compromised nodes. The examples of mission-oriented GCSs include rescue teams with mobile devices for disaster management, soldiers with mobile devices in battlefield situations, mobile robots with embedded sensor systems to explore the surface of Mars [31], and mobile tanks with sensors to survey a hostile battlefield for tracking bio/chemical plumes [31]. In these mission-oriented mobile applications for MANETs, guaranteeing maximum system lifetime while minimizing bandwidth consumed is critical for successful mission execution.

Many emerging mobile applications depend on the notion of secure group communication where mobile nodes can join or leave a group dynamically [6,21,30,36]. A compromised node in a group can compromise the security of the whole system when useful information has been leaked out to the compromised node. Compromised nodes may also collude. To tolerate/detect intrusions, it is essential to dynamically detect and evict compromised nodes by adaptively enhancing the defense in response to the attacker strength. However, if IDS is performed more frequently than needed, it may adversely affect the performance of GCS.

Below we briefly survey existing work in intrusion detection in MANETs and model-based evaluation of intrusion tolerance/detection systems. First, we survey existing techniques (e.g., anomaly-based or misuse-based) and architectures (e.g., hierarchical IDS) used for intrusion detection in MANETs. Second, we survey existing work using model-based evaluation techniques for evaluating intrusion tolerance/detection systems. Also, since our work is to support secure GCSs, we briefly survey basic security properties including availability, confidentiality, integrity, and forward/backward secrecy. Later we will define security failure conditions related to these properties.

While intrusion detection systems (IDS) for wired networks have been extensively studied, there has been little work on IDS for wireless mobile environments, particularly for MANETs. Zhang et al. [42,43] pioneered a distributed and cooperative intrusion detection model based on anomaly detection by which all nodes in the system run IDS to detect and respond to intrusions. Nevertheless, no specific IDS or reactive IDS protocol was discussed. Cluster-based IDS for MANETs has been proposed [2,3,14,15,37,39]. The main idea is that a cluster head (CH) collects security-related information from nodes in the same cluster and determines if any intrusion has occurred. In particular, non-overlapping zone-based IDS was proposed in [38] for MANETs and proven to be effective in intrusion detection. An important issue not addressed is performance degradation due to zone-based IDS. Marti et al. [26] developed a *watchdog* mechanism for identifying misbehaving nodes based on dynamical behaviors and developed a *pathrater* algorithm for routing around misbehaving nodes for

MANETs. Debar et al. [9] suggested aggregation and correlation of IDS alerts to reduce communication/computational overhead caused by performing IDS. There is no tradeoff analysis between security and performance in these studies. No analysis of adaptive IDS in response to growing attacker strength was studied. In this paper, we consider an adaptive IDS design that is reactive to the attacker strength.

Hierarchical IDS was proposed to realize distributed anomaly-based IDS in MANETs [3,4,13,15]. However, the issues of extra latency and energy consumption were not addressed. The assumption that the CH is tamper-resistant and the CH selection process will not be interrupted by attackers is also questionable. Further, only security properties of IDS in MANETs are examined without considering reactive approaches against changes to the attackers' behaviors. Stern et al. [37] proposed data reduction techniques to reduce communication costs in their IDS design. However, detection latency introduced by data aggregation and their effect on performance degradation were not investigated. Our work differs from the prior work in that we consider the impact of IDS on performance degradation, and identify optimal design settings of adaptive IDS based on the tradeoff between performance (i.e., communication cost) versus security (i.e., mean time to security failure or MTTSF) for GCSs in MANETs.

Hasswa et al. [12] and Nadeem et al. [28] proposed IDS to deal with Denial-of-Service (DoS) attacks. Hasswa et al. [12] developed IDS and accordingly a response system called *Routeguard* to improve both network throughput and detection accuracy. Nadeem et al. [28] proposed an anomaly intrusion detection system using chi-sqare test and control chart to detect intrusion and identify intruders to improve throughput as well as to reduce overhead. Santosh et al. [32] and Liu et al. [22] employed game theoretic approaches to detect intrusions and identify anomaly behaviors of nodes in MANETs. There were also studies to investigate systems equipped with both intrusion detection and key management [7,41]. To reduce monitoring overhead for identifying anomaly behaviors of nodes, a method to select critical nodes (e.g., suspicious nodes to monitor) was proposed in [16]. Sen et al. [33] used a learning technique based on artificial intelligence to detect known attacks in MANETs. Our work differs from these prior studies in that we specifically consider the effect of adaptive IDS design on both the performance and security properties of GCSs in MANETs.

Recently, Subhadrabandhu et al. [38] studied the tradeoff between energy/computational/communication resource consumption versus IDS accuracy based on distributed IDS. Algorithms were developed to explore the tradeoff. Our work differs from their work in that we specifically deal with GCSs in MANETs and we allow the optimal design settings of adaptive distributed IDS to be identified, when given the system-imposed security and performance requirements.

To evaluate both security and performance characteristics of IDS in GCSs, the approach used in this paper has its root in model-based quantitative analysis [29]. In the literature, we have seen some recent work extending model-based quantitative analysis to security analysis. Liu et al. [23] and Boppana et al. [1] utilized Markov models to analyze the security properties of a class of intrusion detection algorithms. Liu et al. [23] utilized a Markov decision process to analyze the security characteristics of a framework that combines intrusion detection with authentication for MANETs. Boppana et al. [1] also utilized a Markov model to evaluate false positives of IDS. Dacier et al. [8] proposed a novel approach to model the system as 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. [14] presented a quantitative Markov model of attacker behaviors using data obtained from several experiments conducted over 2 years. They postulated that the process describing an attacker may be divided into multiple phases, such as learning, standard attack,

and innovative attack. Goseva-Popstojanova et al. [11] presented a state transition model to describe dynamic behaviors of intrusion tolerant systems. Their model includes a framework to define the vulnerability and the threat set. Madan et al. [25] employed a semi-Markov process (SMP) model to evaluate security attributes of an intrusion tolerant system. Madan et al. also [25] used a steady-state analysis to obtain dependability measures such as availability, and a transient analysis with absorbing states to obtain security measures such as *mean time to security failure* (MTTSF). Wang et al. [40] utilized a higher-level formalism based on stochastic petri nets (SPN) for security analysis of intrusion tolerant systems. Leversage and James [20] 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 of the previous work cited above, however, often only focused on security measures without considering the impact of deploying security mechanisms on the performance of the system. Further, there was no solution provided to address the system optimization issue in terms of identifying the optimal design settings of adaptive IDS for mission-oriented mobile groups. Our work addresses this issue.

Very recently, Cho and Chen [7] also utilized model-based evaluation techniques to analyze the performance characteristics of IDS based on linear detection integrated with key management based on threshold batch rekeying. However, the focus was on modeling the effect of integrating linear detection-based IDS with threshold-based batch rekeying on the MTTSF of a system in which only a single group exists. In this paper we investigate distributed intrusion detection in general, considering various ways of performing intelligent intrusion detection (logarithm, linear, exponential) in response to attacker strengths. Also unlike the prior work, we consider immediate rekeying for key management as well as network partition/merge possibilities due to node failure and mobility events, such that multiple groups may exist in response to network dynamics. Finally, unlike the prior work, the goal of this paper is to identify the best intrusion detection algorithm (logarithm, linear, exponential), along with the best periodic interval to maximize the MTTSF and to minimize the communication overhead.

The contributions of the paper are as follows. First, we develop model-based evaluation methods to quantitatively analyze the tradeoff between security and performance of mobile GCSs in MANETs in the presence of inside attackers and intrusion detection mechanisms, recognizing the fact that security mechanisms often have impacts on the performance property of the system. We develop an analytical model to succinctly describe the inside attacker, the GCS, and distributed intrusion detection mechanisms with the goal to evaluate the effect of intrusion detection on security and performance properties of the system. Second, when given a set of parameter values characterizing the operational or environmental conditions of the system, we identify the optimal intrusion detection interval under which the MTTSF metric is maximized. Moreover, we effectively trade off security for performance, or vice versa, such that system designers can adjust the intrusion detection interval to maximize MTTSF while satisfying imposed performance requirements in terms of overall communication cost. Third, we propose a robust, efficient, and adaptive distributed intrusion detection mechanism that dynamically adjusts the intrusion detection interval and a detection function optimally reacting to dynamically changing attacker strength. Our IDS protocol considers the effect of possible collusion of compromised nodes as well as IDS intrinsic defects including false positives and false negatives. Lastly, we model our secure GCS for MANETs based on realistic behaviors of inside attackers. To be specific, we consider multiple levels of attacker strength reflecting system conditions and attacker behaviors.

2 Preliminary

2.1 Secure Group Communication Systems in Mobile Ad Hoc Networks

Secure GCSs are most often seen in military settings where combat units spread out in a geographical area without a communication infrastructure but must maintain a consistent view in order to make correct combat decisions. Group members must communicate each other state changes, such as changes of membership of nodes, their locations, and approaching objects. Very typically, such a military deployment is mission-oriented and the goal is to complete the combat mission within system lifetime. In this sense, the security requirement is expressed in terms of a threshold for MTTSF such that the system must be able to survive security threats past the minimum mission time. The timeliness requirement is expressed in terms of the delay requirement per packet. This translates into a maximum network traffic rate which bounds the delay or response time per packet.

An efficient way to achieve secure group communications is to use a symmetric key, called the *group key*, shared by group members. Group members may agree upon the group key by means of a group key agreement protocol in a MANET in which there is no centralized key server. Group members employ the group key to encrypt group messages. By employing the group key as a secret key, only members of the group are able to decrypt and read group messages [21]. This achieves the *confidentiality* property for secure group communications.

In a dynamic group setting where users can join or leave the group at any time, the group key needs to be rekeyed. There are the two main security properties commonly associated with rekeying [30,36], namely, *forward secrecy*, which means that an adversary who knows previous group keys cannot identify subsequent group keys, and *backward secrecy*, which means that an adversary who knows the current group key cannot discover previous group keys. To maintain both backward and forward secrecy, rekeying (i.e., change the group key) is performed whenever a group membership change event occurs due to a user newly joining the group or a current member leaving or being evicted.

For MANETs, there is no centralized trusted key server. Instead a distributed key management protocol, a *contributory key agreement* (CKA) protocol, would be used for rekeying upon a join/leave/eviction event. Without loss of generality, this paper uses a CKA protocol called GDH (Group Diffie-Hellman) [35,36], a well-known distributed key management protocol for secret key generation and distribution in MANETs.

2.2 Distributed Intrusion Detection Protocols

We consider two types of intrusion detection protocols applicable to GCSs in MANETs, i.e., *host-based IDS* versus *voting-based IDS*. The first type is *host-based IDS* [42,43] for local detection in which each node performs local detection to determine if a neighboring node has been compromised. Each node may implement its host-based IDS preinstalled with standard existing IDS techniques such as misuse detection (also called signature-based detection) and anomaly detection [18,19,42,43] so that our proposed voting-based IDS can be independent of the host-based IDS used as a general framework. Each node may evaluate its neighbors based on information collected, mostly route-related and traffic-related information [13]. We measure the effectiveness of IDS techniques applied (e.g., misuse detection or anomaly detection) for host-based IDS by two parameters, namely, the false negative probability ($p1$) and false positive probability ($p2$). In general, when the system uses misuse detection for

IDS, it tends to have more false negatives and less false positives (e.g., higher *p1* and lower *p2*). On the other hand, when the system employs anomaly detection for IDS, it is likely to have fewer false negatives and more false positives (e.g., lower *p1* and higher *p2*).

The second type is *voting-based IDS* for cooperative detection based on majority voting. Voting has been used as a mechanism to achieve fault tolerance [5]; we adopt it to cope with intrusions. For voting-based IDS to be performed, each node is preinstalled with host-based IDS to collect information to detect the status of neighboring nodes. Periodically a target node would be evaluated by m vote-participants dynamically selected where m is a design parameter. If the majority of m nodes decided to vote against the target node, then the target node would be evicted from the system by means of rekeying. This adds intrusion tolerance to tolerate collusion of compromised nodes in MANETs as it takes the majority of "bad" nodes among m nodes to work against the system. We characterize voting-based IDS by two parameters, namely, false negative probability (P_{fn}) and false positive probability (P_{fp}). These two parameters can be calculated based on (a) the *per-node* false negative and positive probabilities (*p1* and *p2*) of host-based IDS in each node; (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., $\geq \lceil m/2 \rceil$) cast negative votes against a target node, the target node is regarded as compromised and will be evicted from the system.

3 System Model

This paper concerns a mission-oriented GCS consisting of mobile groups in MANETs equipped with intrusion detection to mainly deal with inside attackers. We define a mobile group based on "connectivity." When all nodes are connected, there is only a single group in the system. That is, group members must maintain connectivity for them to be in the same group. The GCS and its constituent mobile groups are "mission-oriented" in the sense that a mobile group may be partitioned into several groups due to network partition derived from node mobility or node failure, but each group will continue to execute the mission amid group partition and merge activities. Moreover, the GCS fails when any mobile group fails, modeling the case in which a security failure of any mobile group compromises the mission assigned, e.g., in secret mission situations. We assume that each member has a private key and its certified public key available for *authentication* purposes. When a new member joins a mobile group, the new member's identity is authenticated based on the member public/private key pair by applying the challenge/response mechanism. We assume that the GCS maintains *view synchrony* (VS) [6,21,30,36] which guarantees that messages are delivered reliably and in order. We characterize the workload and operational conditions of a GCS in MANETs by a set of model parameters. We assume that the inter-arrival times of join and leave requests are exponentially distributed with their rates being λ and μ respectively. Also we assume that the inter-arrival time of data packets issued by a node for group communication is exponentially distributed with rate λ_q. The assumption of exponential distribution can be relaxed since the SPN performance model developed is capable of allowing any general distribution for a transition time. We assume that the time to perform a rekeying operation upon a membership change event (i.e., join or leave event) or a forced eviction is measured based on GDH [35,36] to realize distributed key management in MANETs.

3.1 Attacker Model, IDS Accuracy, and Security Failure Definition

Secure GCSs in MANETs must meet four requirements in the presence of inside and outside attackers: confidentiality, integrity, availability, and authentication. The availability property is achieved by maximizing the lifetime of the system in the presence of insider attacks. The other security properties can be achieved by using intrusion prevention techniques, such as encryption (i.e., hash functions), authentication (MAC-message authentication code or public/private key pairs preinstalled), to deal with outsider attacks.

We assume that there are intrusion prevention techniques in place, such as encryption or authentication, to deal with outsider attacks (e.g., disrupting traffic, modifying data, eavesdropping). Insider attacks are due to compromised nodes disguised as legitimate members to disrupt the system. An inside attacker may obtain secret information and pass it to outside attackers (i.e., illegal data leak out) to compromise the system. It can also collude with other inside attackers to compromise other good nodes or entice the system to evict good nodes from the system. When the system is populated with too many compromised nodes, a security failure will also occur. In the paper, we will use the terms inside attackers (or simply attackers) and compromised nodes interchangeably.

Recognizing the principles in [10] that attacker behaviors are not always random, we use three attacker functions to model the attacker strength based on the prediction of time and effort made to perform an attack as follows:

- *Logarithmic time attacker*: The attacker increasingly takes longer time to compromise nodes in the system, following a logarithmic function curve. This models the scenario where the system has detected attackers (i.e., compromised nodes) and enhanced the defenses of the remaining nodes, making it increasingly harder for the attacker to compromise more nodes.
- *Linear time attacker*: The attacker compromises nodes one after the other with the node compromising rate linear to the number of compromised nodes in the system. This applies to the case in which compromised nodes do not collude and just perform constant time attacks.
- *Polynomial time attacker*: The attacker increasingly takes shorter time to compromise nodes in the system, following an exponential function curve. This models the scenario where the attacker learns secret information from compromised nodes in the system and exploits it to more easily compromise other nodes within a shorter time.

We assume that IDS will perform its function periodically. The detection interval is dynamically adjusted in response to the accumulated number of intrusions that have been detected in the system. Similar to the attacker behavior model above, we consider three detection functions to model the IDS activities in terms of periodicity, namely, *logarithmic periodic detection, linear periodic detection*, and *polynomial periodic detection*, as follows:

- *Logarithmic periodic detection*: In this detection scheme, the system performs intrusion detection in a conservative way with a rate logarithmic to the number of compromised nodes that have been identified. This detection approach is usually applicable in a low or moderate level of hostile network environments. Further, this can be effective to save energy consumption introduced by IDS as well as to reduce false positives.
- *Linear periodic detection*: This system performs IDS with a linear rate to the number of intrusions that have been detected in the system. Since this approach performs IDS more frequently than logarithmic periodic detection, it is a suitable detection approach when the employed IDS technique has high accuracy with relatively low number of false positives and negatives.

- *Polynomial periodic detection*: This detection scheme aggressively performs IDS by increasing the detection rate in a polynomial fashion to the number of observed compromised nodes in the system. This scheme is effective for very high quality IDS with very low false negatives and false positives. Otherwise, using this scheme has the adverse effect of having more false positives and negatives by performing IDS more than it needs.

To realize a fully distributed intrusion detection mechanism in MANETs, we assume that each node has host-based IDS preinstalled to perform intrusion detection activities. Host-based IDS has two key parameters, *p1* and *p2*, characterizing the false negative probability and false positive probability respectively related to the accuracy or quality of IDS. Each node casts a vote for or against a target node based on the decision made using its host-based IDS. To alleviate collusion among compromised nodes, the system could perform voting-based IDS by which the majority of vote-participants must agree to evict a target node before the target node is evicted. The number of vote-participants (m) is a system parameter whose effect will be analyzed in the paper. Voting-based IDS is characterized by the false negative probability (P_{fn}) and false positive probability (P_{fp}) which depend on *p1* and *p2*, respectively, and the number of compromised nodes in the group. We define two security group failure conditions so that a mobile group enters a security failure state when one of the two security group failure conditions stated below is true. That is, the GCS fails if any of mobile groups fails when either Condition C1 or C2 listed below is true.

- *Condition C1*: a compromised but undetected member requests and subsequently obtains data using the group key. The system is in a failure state because data have been leaked out to a compromised node, leading the *loss of system integrity* in a security sense.
- *Condition C2*: more than 1/3 of member nodes are compromised but undetected by IDS. We assume the *Byzantine Failure* model [10] such that when more than 1/3 of member nodes in a mobile group are compromised, the mobile group is compromised, resulting in the *loss of availability* [17] of system service.

If a member node is detected as compromised by IDS, the system won't allow the member node to request data anymore and will evict the member immediately to satisfy the forward/backward secrecy requirement, one of confidentiality properties. After a node is detected as compromised and evicted from the system, it cannot rejoin the group again. That is, there is no recovery mechanism available in the system to repair a compromised member and make it a trusted member node again. Initially, all nodes are trusted.

3.2 Security and Performance Metrics

- *MTTSF (mean time to security failure)*: This metric indicates the average time elapsed for the GCS to reach a security failure state. The GCS fails when any mobile group reaches a security failure state when (1) data have been leaked out to a compromised but undetected member node (i.e., C1), or (2) more than 1/3 of the member nodes have been compromised (i.e., C2). Note that illegal data leak out only occurs when a compromised but undetected member requests data and subsequently obtains data using the group key. As a security metric, lower MTTSF means faster *loss of system integrity* or *loss of availability*. A design goal is to maximize MTTSF.
- *Communication traffic cost* ($\hat{C}_{total}$): This metric indicates total traffics incurred per time unit (sec) including group communication, status exchange, rekeying, intrusion detection, beacon, group partition/merge and mobility-induced activities. Since all nodes share a

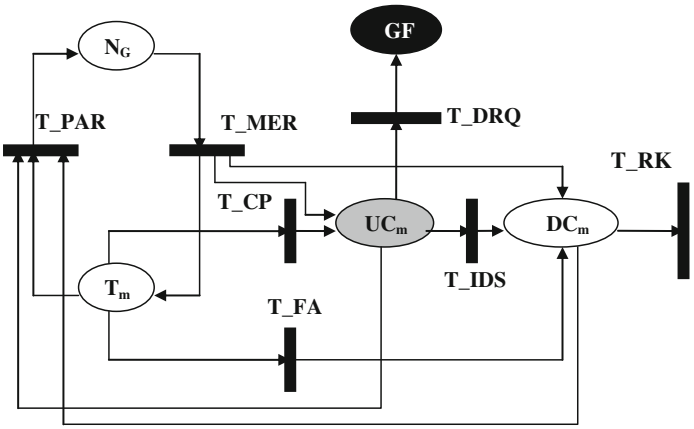


Fig. 1 SPN performance model

Table 1 Places, transitions, and transition rates for the SPN model in Fig. 1

Place	Meaning
T_m	Mark(T_m) means the number of trusted member nodes
UC$_m$	Mark(UC$_m$) means the number of compromised but undetected member nodes
DC$_m$	Mark(DC$_m$) means the number of compromised and detected member nodes
GF	Mark(GF) $= 1$ means that group security failure has occurred due to illegal data leak-out

Transition	Rate	Physical meaning
T_CP	$A(m_c)$	A node has been compromised
T_IDS	mark(UC$_m$) $* D(m_d) * (1 - P_{\text{fn}})$	A compromised node has been detected
T_FA	mark$(T_m) * D(m_d) * P_{\text{fp}}$	A node has been falsely diagnosed as compromised
T_RK	$1/T_{cm}$	A rekeying operation has been performed
T_DRQ	mark(UC$_m$) $* p1 * \lambda_q$	A group communication operation has been performed

wireless bandwidth BW, a high $\hat{C}_{\text{total}}$ will be translated into a high level of contention and consequently a high delay or response time for group communication. A design goal is to minimize $\hat{C}_{\text{total}}$.

4 Performance Model

We develop a stochastic petri net (SPN) model as shown in Fig. 1 to describe the behavior of a mobile group in the presence of insider attacks and intrusion detection activities, with the goal of identifying optimal design settings (i.e., optimal intrusion detection interval, detection function, and number of vote-participants) to maximize MTTSF while satisfying imposed performance (i.e., overall communication cost) requirements. Table 1 summarizes places and transitions along with transition rates in the SPN model with their physical meanings given.

Below we describe how the SPN model is constructed:

- The SPN model describes the behavior of a *single* mobile group as it evolves. This mobile group may partition into two and may merge with another group during its lifetime. We track trusted members, compromised members undetected, and compromised members detected during the group's lifetime to understand its security and performance characteristics. Certainly the system knows the number of compromised nodes detected by IDS at all times. However, the system does not know the number of compromised nodes not yet detected. It only knows the total number of member nodes. The SPN model predicts the number of compromised but yet detected nodes through knowledge of the node compromising rate or the attacker function explained below.
- 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. To be more specific, the numbers of nodes in places T_m, UC_m, and DC_m, obtained by $\mathrm{mark}(T_m)$, $\mathrm{mark}(\mathrm{UC}_m)$, and $\mathrm{mark}(\mathrm{DC}_m)$, respectively, would be adjusted based on the number of groups existing in the system (obtained by mark (N_G)), which changes upon a group merge/partition event.
- We use transitions to model events. Specifically, T_MER and T_PAR model group merge and partition events, respectively; T_CP models a node being compromised. T_FA models a node being falsely identified as compromised. T_IDS models a compromised node being detected. T_RK models rekeying. T_DRQ models a data leak security failure (i.e., C1). 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, $\mathrm{mark}(N_G)$ changes upon firing T_MER or T_PAR since the number of groups changes upon a group merge or partition event; the number of compromised nodes undetected increments by 1 and, 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. The firing conditions are (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 exists "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. See Eq. 3 for the parameterization of $A(m_c)$.
- 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., C1. A compromised and undetected member will attempt to compromise data from other members in the group. Because of the use of host-based IDS, a node will reply to such a request only if it could not identify the requesting node as compromised with the false negative probability $p1$. This is modeled by associating transition T_DRQ with rate $p1 * \lambda_q * \mathrm{mark}(\mathrm{UC}_m)$. The firing of transition T_DRQ will move a token into place GF, at which point we regard the system as experiencing a security failure due to 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 mobile group is modeled by the IDS detection rate $D(m_d)$. See Eq. 4 for the parameterization of $D(m_d)$. 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 but undetected node now becomes detected by IDS. For voting-based IDS, the transition rate of T_IDS is $\text{mark}(UC_m) * D(m_d) * (1 - P_{\text{fn}})$, taking into consideration of the number of compromised but yet detected nodes and the false negative probability of voting-based IDS. 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 $\text{mark}(T_m) * D(m_d) * P_{\text{fp}}$. Note that 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 mobile group experiences a security failure if either security failure condition, C1 or C2, is met. We model this by making the group enter an absorbing state when either C1 or C2 is *true*. To achieve this, we associate every transition in the SPN model with an enabling function that returns *false* (disabling the transition from firing) when either C1 or C2 is met, and returns *true* otherwise. For the SPN model, C1 is *true* when mark (GF) > 0 representing that data have been leaked out to compromised, undetected members; C2 is *true* when more than 1/3 of member nodes are compromised but undetected as indicated by:

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

4.1 Parameterization

Here we describe the parameterization process, i.e., how to give model parameters proper values reflecting the operational and environmental conditions of the system.

- T_{cm}: Recall that T_{cm} is the communication time required for broadcasting a rekey message for a join or leave event. The reciprocal of T_{cm} is the rate of transition T_RK. Based on GDH, the following formula is used to calculate T_{cm}:

$$if\ (N > 1)\ T_{\text{cm}} = \frac{3b_{\text{GDH}}\,(N-1)}{\text{BW}} \quad \text{else} \quad T_{\text{cm}} = \frac{b_{\text{GDH}}}{\text{BW}} \tag{2}$$

where N is the number of *current* member nodes indicated by $\text{mark}(T_m) + \text{mark}(UC_m)$, b_{GDH} is the length of an intermediate value, and BW is the wireless network bandwidth (Mbps). We assume that the size of the rekey message is at least b_{GDH} when the current number of members is zero or one.

- $A(m_c)$: This is an attacker function that returns the rate at which nodes are compromised in the mobile group. It will apply to transition T_CP in the SPN model. Three different attacker strengths are considered based on the time taken to compromise a node, namely, *logarithmic*, *linear*, and *polynomial time attacker,* as follows:

$$A_{\log}(m_c) = \lambda_c \times \log_p(m_c), \quad A_{\text{linear}}(m_c) = \lambda_c \times m_c,$$
$$A_{\text{poly}}(m_c) = \lambda_c \times (m_c)^p \quad \text{where} \quad m_c = \frac{\text{mark}(T_m) + \text{mark}(\text{UC}_m)}{\text{mark}(T_m)} \tag{3}$$

These three attacker strengths differ by the way the node compromising rate increases as more nodes become compromised. For the linear attacker function, the node compromised rate increases linearly with the number of compromised nodes. Hence, $A_{\text{linear}}(m_c) = \lambda_c m_c$ where m_c reflects the degree of compromised nodes currently in the group and λ_c is the base node compromising rate initially given that there is no compromised node in the group. For $A_{log}(m_c)$, the compromising rate increases in logarithm form with the number of compromised nodes. For $A_{\text{poly}}(m_c)$ the compromising rate increases in exponential form with the number of compromised nodes. Note that these three forms are prediction functions for the node compromising rate. The base compromising rate (λ_c) can be obtained by first-order approximation from observing the number of compromised nodes over a time period. We also note that p is a base index parameter selected to reflect the degree of changes of the logarithmic and polynomial attacker functions with respect to the number of compromised nodes. It requires fine tuning after sufficient data are collected. We choose $p = 3$ in this paper.

- $D(m_d)$: This is a detection function that returns the rate at which IDS is invoked. Three different detection functions, namely, *logarithmic*, *linear*, and *polynomial periodic detection,* are parameterized as follows:

$$D_{\log}(m_d) = \frac{1}{T_{\text{IDS}}} \times \log_p(m_d), D_{\text{linear}}(m_d) = \frac{1}{T_{\text{IDS}}} \times m_d,$$
$$D_{\text{poly}}(m_d) = \frac{1}{T_{\text{IDS}}} \times (m_d)^p \quad \text{where} \quad m_d = \frac{N_{\text{init}}}{\text{mark}(T_m) + \text{mark}(\text{UC}_m)} \tag{4}$$

These three functions differ by the way the detection rate changes with the number of compromised nodes that have been detected by IDS. For the linear detection function, the IDS detection rate increases linearly with the number of compromised nodes detected. $D_{\text{linear}}(m_c)$ is the linear periodic detection function where m_c indicates the degree of compromised nodes that have been detected by IDS, and T_{IDS} is the base detection time interval which we aim to determine for maximizing MTTSF when applying voting-based IDS. The logarithmic detection function, $D_{\log}(m_d)$, and exponential detection function, $D_{\text{poly}}(m_d)$, have the same form as their counterparts in the attacker function. Note that p is defined similarly as in Eq. 3.

- *Group merge and partition*: We model group merge and partition events by a *birth-death process* with arrival rate = $\lambda_{np,i}$ and departure rate = $\mu_{nm,i}$ where state i represents that there are i mobile groups in the GCS. We obtain *group* merging/partitioning rates via simulation. The assumptions used are: (1) a random way point (RWP) mobility model is used to describe a node's mobility behavior; and (2) the inter-arrival time of a node's join/leave operation is exponentially distributed. 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 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, the merging/partitioning rates in state i, represented by $\mu_{nm,i}$ and $\lambda_{np,i}$, are computed by the first-order approximation as:

$$\mu_{nm,i} = \frac{N_{nm,i}}{S_i}, \qquad \lambda_{np,i} = \frac{N_{np,i}}{S_i} \tag{5}$$

Note that the group merging/partitioning rates parameterized above are functions 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, leading to a smaller number of groups (lower i) observed in the system. On the other hand, as the node density is low, the system is more likely to stay at large number of groups (higher i) with high probability. In other words, when the node density is low, group partition is more likely to occur than group merge.

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

$$\text{where} \quad N_{\text{majority}} = \left\lceil \frac{m}{2} \right\rceil, N = N_{\text{good}} + N_{\text{bad}} = \text{mark}\,(T_m) + \text{mark}\,(\text{UC}_m) \quad \text{and}$$

$$C\begin{pmatrix} n \\ k \end{pmatrix} = 0 \quad if \quad n < k. \tag{6}$$

- P_{fn}, P_{fp}: P_{fn} is the probability of false negatives defined as the number of compromised nodes diagnosed by voting-based IDS 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 the probability of false positives defined as the number of normal nodes flagged as anomaly over the number of trusted normal nodes. We consider the intrinsic defect of host-based IDS in each node as well as the possible collusion of compromised nodes during the voting process. If a vote-participant is compromised, it can cast a negative vote to evict a healthy target node in the group or it can cast a positive vote for a malicious node to keep more compromised nodes in the group. Equation 6 reflects these two cases of false positives or false negatives introduced into the group respectively. Here m is the number of vote-participants to cast a vote against a target node, p_{f} is $p1$ for calculating P_{fn} *or* $p2$ for calculating P_{fp}, N_{bad} is the number of currently compromised nodes in the group represented as mark(UC_m), and N_{good} is the number of currently healthy nodes in the group indicated as mark(T_m). P_{fp} is obtained when the majority of voters consists of bad nodes who cast a negative vote against a good node, and good nodes who mistakenly diagnose a good node as a bad node with the probability of $p2$ (i.e., $p2$ is a per-node false positive probability), resulting in a healthy node being evicted. On the other hand, P_{fn} occurs when the majority of voters is from positive votes by bad nodes (i.e., casting a positive vote against a bad node) or good nodes who mistakenly diagnose a bad node as a good node with the probability of $p1$ (i.e., $p1$ is a per-node false negative probability), keeping more compromised nodes undetected in the group. Note that for given $p1$ and $p2$ values, P_{fn} and P_{fp} still vary dynamically, reflecting changing network and operational conditions, such as the degree of compromised nodes, node density, and number of vote-participants (m) used over time.

4.2 MTTSF and $\hat{C}_{\text{total}}$ Calculation

MTTSF is obtained using the concept of *mean time to absorption* (MTTA) 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 C1 or C2 is met, the system fails). Then the MTTA or the MTTSF of the system is simply the expected accumulated reward until absorption, $E[Y(\infty)]$, defined as:

$$E[Y(\infty)] = \sum_{i \in s}^{\infty} r_i \int_0^{\infty} P_i(t)\, dt \tag{7}$$

where S denotes the set of all states except the absorbing states, r_i (reward) is 1 for those states, and $P_i(t)$ is the probability of state i at time t.

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

$$\hat{C}_{\text{total}} = \frac{\int_0^{\text{MTTSF}} \hat{C}_{\text{total},i}(t)\, dt}{\text{MTTSF}} \tag{8}$$

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

$$\hat{C}_{\text{total},i} = \hat{C}_{\text{GC},i} + \hat{C}_{\text{status},i} + \hat{C}_{\text{rekey},i} + \hat{C}_{\text{IDS},i} + \hat{C}_{\text{becon},i} + \hat{C}_{\text{mp},i} \tag{9}$$

where $\hat{C}_{\text{GC,i}}$, $\hat{C}_{\text{status,i}}$, $\hat{C}_{\text{rekey,i}}$, $\hat{C}_{\text{IDS,i}}$, $\hat{C}_{\text{beacon,i}}$, and $\hat{C}_{\text{mp,i}}$, are the costs per time unit 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. Below we explain how we parametrize $\hat{C}_{\text{GC,i}}$, $\hat{C}_{\text{status,i}}$, $\hat{C}_{\text{rekey,i}}$, $\hat{C}_{\text{IDS,i}}$, $\hat{C}_{\text{beacon,i}}$ and $\hat{C}_{\text{mp,i}}$. Note that we have omitted (t) in each term of Eq. 9 for simplicity.

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

$$\hat{C}_{\text{GC},i} = \lambda_q \times N \times b_{\text{GC}} \times H_i \tag{10}$$

where λ_q is the group communication rate, N is the number of active group members in the single group we are observing (i.e., $\text{mark}(\text{UC}_m) + \text{mark}(T_m)$), b_{GC} is the message size (bits) of a group communication packet, and H_i is the number of hops a multicast packet travels from a node to all group members connected by a binary tree structure, given as:

$$H_i = r_i / R \times (N - 1) \quad \text{where} \quad r_i = r/\sqrt{i} \tag{11}$$

Here r_i is the radius of the operational group area when i groups exist in the system and R is the wireless radio range used.

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

$$\hat{C}_{\text{status},i} = \frac{(N \times b_s) \times H_i}{T_{\text{status}}} \tag{12}$$

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

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

$$\hat{C}_{\text{rekey},i} = \hat{C}_{\text{join/leave},i} + \hat{C}_{\text{eviction},i} \tag{13}$$

where $\hat{C}_{\text{join/leave},i}$ is the cost introduced by join and leave operations per time unit and $\hat{C}_{\text{eviction},i}$ is the cost introduced by forced evictions per time unit. The term $\hat{C}_{\text{join/leave},i}$ is calculated by:

$$\hat{C}_{\text{join/leave},i} = \Lambda_J \times C_{\text{join},i} + \Lambda_L \times C_{\text{leave},i} \tag{14}$$

where Λ_J and Λ_L are aggregate join and leave rates with $\Lambda_J = \lambda \times N \times \mu/(\lambda + \mu)$ and $\Lambda_L = \mu \times N \times \lambda/(\lambda + \mu)$, and $C_{join,i}$ and $C_{leave,i}$ are the rekeying costs per join/leave operation, calculated as:

$$C_{\text{join},i} = C_{\text{leave},i} = (H_i \times M_{\text{updates}}^{\text{members}}) + C_{\text{GDH},i} \tag{15}$$

$$C_{\text{GDH},i} = \{b_{\text{GDH}}(2N - 3) \times (N - 1)\} + \{b_{\text{GDH}} \times N \times H_i\} \tag{16}$$

Here H_i is as given in Eq. 11, $M_{\text{update}}^{\text{members}}$ is the number of bits to update the group member view, and $C_{\text{GDH},i}$ is the rekeying cost when i groups exist in the system. In Eq. 16, the first term indicates the unicast communication cost in stages of 1 and 3 of GDH while the second term accounts for the multicast communication cost in stages of 2 and 4 of GDH. The term $\hat{C}_{\text{eviction},i}$ in Eq. 13 is calculated as:

$$\hat{C}_{\text{eviction},i} = [\text{rate}(T\_\text{IDS}) + \text{rate}(T\_\text{FA})] \times \hat{C}_{\text{leave},i} \tag{17}$$

where rate(T_IDS) is the IDS intrusion detection rate and rate(T_FA) is the IDS false alarm rate by which nodes are identified as compromised nodes. Both rates may be obtained readily from evaluating the SPN performance model.

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

$$\hat{C}_{\text{IDS},i} = D(m_d) \times (1 - P_{\text{fn}}) \times N \times [b_{m-\text{list}} + m \times b_v] \times H_i \tag{18}$$

where $D(m_d)$ is the detection rate, 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, $b_{m-\text{list}}$ is the message size (bits) of the list containing m vote participants, and b_v is the message size (bits) of a vote.

- $\hat{C}_{\text{beacon},i}$: this is the communication cost due to beaconing messages being multicast to group members. It is calculated by:

$$\hat{C}_{\text{beacon},i} = \Lambda_{\text{RB}} \times M_{\text{alive}} \times H_i \tag{19}$$

$$\Lambda_{\text{RB}} = N \times \left\lceil \frac{N}{\lambda + \mu} \right\rceil \times \frac{1}{T_{\text{RB}}} \tag{20}$$

- $\hat{C}_{mp,i}$: this is the communication cost due to group merge and partition events. It is computed by:

$$\hat{C}_{mp,i} = C_{\text{partition},i} + C_{\text{merge},i} \tag{21}$$

$$C_{\text{partition},i} = \lambda_{np,i} \times C_{np,i} \tag{22}$$

$$C_{\text{merge},i} = \mu_{nm,i} \times C_{nm,i} \tag{23}$$

$$C_{np,i} = 2 \times (H_i \times M_{\text{update}}^{\text{members}} + C_{\text{rekey},i}), \qquad C_{nm,i} = H_i \times M_{\text{update}}^{\text{members}} + C_{\text{rekey,i}} \tag{24}$$

Table 2 Parameters and their default values

Parameter	Values	Parameter	Values	Parameter	Values
λ	once per 1 h	$p1 = p2$	1%	b_v	8 bytes
μ	once per 4 h	r	500 m	b_{GC}	100 bytes
T_{IDS}	5–1,200 s	N_{init}	100	b_{GDH}	8 bytes
T_{status}	2 s	$D(m_d)$	Linear attack	p	3
λ_c	once per 12 h	$A(m_c)$	Linear detection	m	5
λ_q	once per min	b_s	50 bytes	BW	1 Mbps

where $\lambda_{np,i}$ and $\mu_{nm,i}$ are group partitioning and merging rates where there exist i groups in the system, as given in Eq. 5, $C_{np,i}$ and $C_{nm,i}$ are communication costs generated by group partition and merge events when the system is in state i, $M_{\text{update}}^{\text{members}}$ is the number of bits to update the group member view, and H_i is the number of hops from a node to other group members as given in Eq. 11.

5 Numerical Data and Analysis

We present numerical data obtained through the evaluation of the SPN model developed and provide physical interpretations. Our objective is to identify the optimal intrusion detection interval (T_{IDS}) that will maximize MTTSF while satisfying performance requirements of the system. We also identify the best detection function to use in response to the attacker function (compromising rate) detected at runtime. Extensive simulation has been conducted to validate analytical results obtained.

Table 2 summarizes the set of parameters and their default values used. In particular, the bandwidth of 1 Mbps is a reasonable assumption for a node in MANETs based on 802.11, especially in geographic locations with physical obstacles; $p1$ and $p2$ are selected at 1% based on the assumption of medium to high-quality IDS being used; and, the node speed is derived from the mobility rate, as given by Eq. 25 below. We note that $p1$ and $p2$ are two parameters representing false negative and false positive probabilities for characterizing any host-based IDS. We vary the values of key parameters including the number of vote-participants in voting-based IDS (m), group communication rate (λ_q), and base compromising rate (λ_c) to analyze their effects on the optimal base detection interval for maximizing MTTSF. The ratio of join to leave events is set to 4, reflecting the fact that nodes join a group much faster than they leave a group. Group members communicate with other group members once per 2 min. The rate at which nodes are compromised is once per 12 h, reflecting a medium-high level of attack strength by the attackers.

5.1 Analysis

We first analyze the effect of intrusion detection interval (T_{IDS}) on MTTSF as a function of the number of vote-participants (m) and demonstrate that there exists an optimal intrusion detection interval (T_{IDS}) for maximizing MTTSF or minimizing $\hat{C}_{\text{total}}$ with proper physical interpretations given.

Figure 2 shows the effect of an intrusion detection interval (T_{IDS}) on MTTSF as the number of vote-participants (m) changes for the case in which the attacker function and the

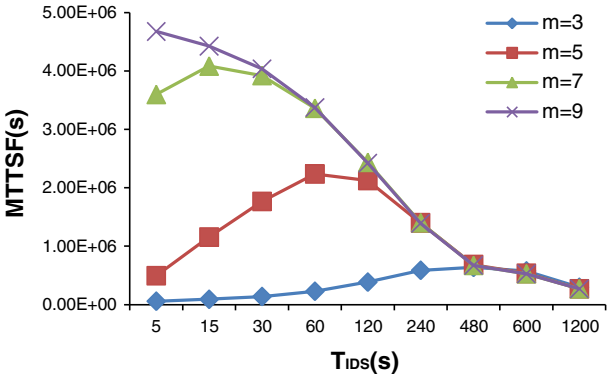


Fig. 2 Effect of m on MTTSF and optimal T_{IDS}

detection function are both linear. We observe that there exists an optimal T_{IDS} that maximizes MTTSF for each given m value. In general, as T_{IDS} becomes larger, MTTSF increases until its optimal point reaches, and then MTTSF decreases after the optimal point. The reason of increasing MTTSF as T_{IDS} increases initially is that as T_{IDS} increases there are fewer nodes being falsely identified by IDS since IDS is triggered less often, thus reducing the system failure probability due to C2. Here we note that P_{fp} is one aspect of false alarms generated by IDS, and therefore more nodes will be falsely identified as compromised nodes if IDS is frequently triggered. After the optimal T_{IDS} is reached, MTTSF decreases again as T_{IDS} increases because IDS is not triggered often enough to detect compromised nodes which may perform attacks to cause system failures due to C1.

We also observe the sensitivity of optimal T_{IDS} identified on MTTSF as m varies. When m is large, the false alarm probability ($P_{\text{fp}} + P_{\text{fn}}$) is small because more nodes are participating in the voting process, thereby reducing the possibility of collusion by compromised nodes. Consequently, when m is large, we observe a high MTTSF due to the small false alarm probability. Conversely, when m is small, MTTSF is small due to a larger false alarm probability. A smaller m also results in a longer optimal T_{IDS} being used to maximize MTTSF to offset the adverse effect of IDS with large false positives, e.g., optimal $T_{\text{IDS}} = 480$, 60, 15, and 5 s for $m = 3$, 5, 7, and 9 respectively.

Figure 3 shows the overall communication cost ($\hat{C}_{\text{total}}$) versus intrusion detection interval (T_{IDS}) as the number of vote-participants (m) varies. An optimal T_{IDS} exists in each curve (minimum $\hat{C}_{\text{total}}$) because of the tradeoff between decreasing normal group communication costs ($\hat{C}_{\text{GC},i}$) and increasing IDS related communication costs ($\hat{C}_{\text{eviction},i} + \hat{C}_{\text{IDS},i}$) as T_{IDS} becomes shorter. Also we observe that when m is large, $\hat{C}_{\text{total}}$ is high. This is because a larger m induces a lower P_{fp} under which more nodes will be able to perform normal group activities. Furthermore, when there are more vote participants, there is a higher cost associated with dynamic majority voting. Contrary to MTTSF versus T_{IDS} in Fig. 2, we do not observe the sensitivity of an optimal T_{IDS} identified, but there is a relatively higher communication cost saved when the optimal T_{IDS} identified is employed as m increases.

Next we analyze the effect of detection functions $D(m_d)$ on MTTSF. Also as an example of applicability, we investigate how one can select the best detection interval (T_{IDS}) and detection function $D(m_d)$ to optimize MTTSF while satisfying the performance requirement in terms of communication overhead, when given the attacker function $A(m_c)$ detected at runtime.

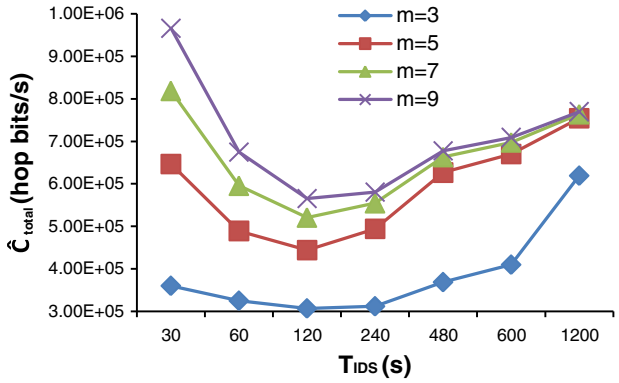


Fig. 3 Effect of m on $\hat{C}_{\text{total}}$ and Optimal T_{IDS}

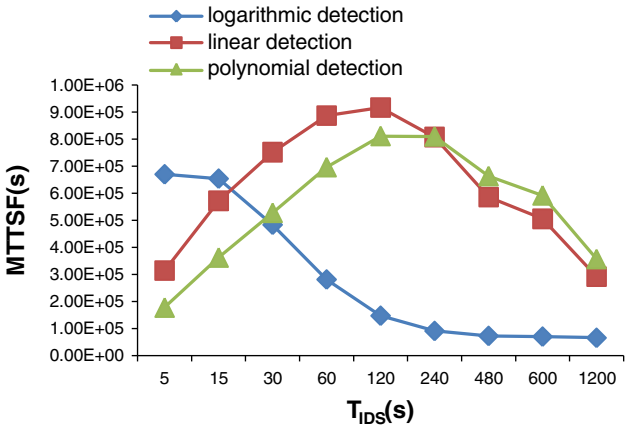


Fig. 4 Effect of T_{IDS} on MTTSF with respect to $D(m_d)$ under linear time attacker function when $m = 5$

In Fig. 4, we show MTTSF versus T_{IDS} for the three detection functions $D(m_d)$ given that the attacker function is linear. We see that each curve again has its own optimal T_{IDS}. The linear detection function $D_{\text{linear}}(m_d)$ shows the best performance at $T_{\text{IDS}} = 120$ s generating the highest MTTSF overall, while the logarithmic detection function $D_{log}(m_d)$ is the worst, particularly when T_{IDS} is sufficiently small. This tradeoff is attributed to the speed of detection (log, linear, or exponential) versus the speed of attack (linear). If the former is greater than the latter, many false positives may be generated; conversely, many compromised nodes may remain in the system. The linear detection function matches up with the linear attacker function the best among the three detection functions in terms of the tradeoff of the two ends. With similar reasoning, we see that the strongest polynomial detection function $D_{\text{poly}}(m_d)$ performs the best for a large T_{IDS} (e.g., $T_{\text{IDS}} > 240$ s) while the weakest logarithmic detection function $D_{log}(m_d)$ performs the best for a small T_{IDS} ($T_{\text{IDS}} < 15$ s).

Figure 5 shows the overall communication cost ($\hat{C}_{\text{total}}$) versus T_{IDS} for the three detection functions $D(m_d)$ given that the attacker function is linear. Each curve in Fig. 5 also has an optimal T_{IDS} that minimizes $\hat{C}_{\text{total}}$. The general trend of the optimal T_{IDS} identified is similar to that shown in Fig. 4 although the exact optimal T_{IDS} points identified are different. The best

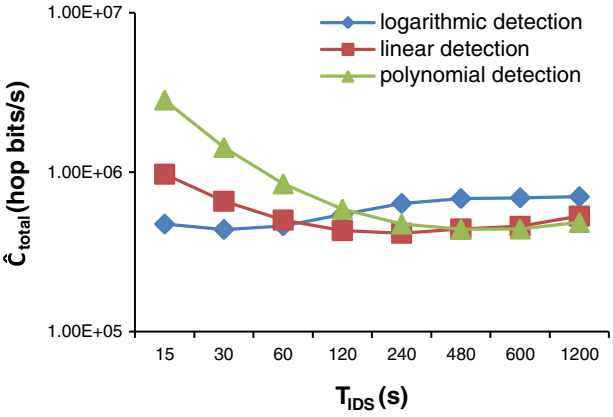


Fig. 5 Effect of T_{IDS} on $\hat{C}_{\text{total}}$ with respect to $D(m_d)$ under linear time attacker function when $m = 5$

performance of $\hat{C}_{\text{total}}$ is observed with linear detection at $T_{\text{IDS}} = 240$ s while the worst performance of $\hat{C}_{\text{total}}$ is shown with logarithmic detection under the ranges of $T_{\text{IDS}} > 120$ s and with polynomial detection under the ranges of $T_{\text{IDS}} \leq 120$ s, resulting in the best performance with linear detection overall. Also in terms of the optimal T_{IDS} identified to minimize $\hat{C}_{\text{total}}$, we see that a shorter optimal T_{IDS} is preferred with less aggressive logarithmic detection, since a shorter T_{IDS} contributes to nodes being evicted more often, consequently leading to less group communication activities. On the other hand, as the detection function becomes aggressive, i.e., polynomial detection, a longer optimal T_{IDS} is favorable to minimize $\hat{C}_{\text{total}}$ in order not to increase too much IDS related traffic more than needed due to aggressive IDS.

In Figs. 6 and 7, we exemplify the selection of optimal design settings in terms of the intrusion detection interval (T_{IDS}), the number of vote-participants (m), and the IDS detection function $D(m_d)$, once we identify the type of attacker function and performance constraints set by the GCS at runtime. Fig. 6 shows the optimal MTTSF and the associated $\hat{C}_{\text{total}}$ values obtainable under three detection functions (log, linear and exponential), when given the type of attacker function (in the X coordinate) and the performance constraints set by the GCS system, i.e., $\text{MTTSF}_{\text{TH}} = 600{,}000$ s and $\hat{C}_{\text{total,TH}} = 500{,}000$ hop bits/s representing less stringent performance constraints. Table 3 lists the actual optimal values obtained as well as the optimal settings (m, T_{IDS}) under which the optimal values are obtained. Suppose that due to the criticality of mission-oriented applications in MANETs, the design goal is to maximize MTTSF while satisfying performance constraints. One can then do a table lookup to select the linear IDS detection function to achieve the goal, given that a particular attack function has been detected at runtime. Specifically, from Table 3 one would select to apply the setting of $m = 9$ and $T_{\text{IDS}} = 60$ s for logarithmic attacks, and $m = 7$ and $T_{\text{IDS}} = 120$ s for both linear and polynomial attacks. This table lookup can be performed upon detection of the attacker function from observing historical data on compromised nodes that have been detected by IDS.

Figure 6 is for the scenario in which performance constraints are relatively less stringent where all three detection functions satisfy the imposed performance constraints. Figure 7 shows the optimal MTTSF and the associated $\hat{C}_{\text{total}}$ values obtainable for a scenario in which performance constraints imposed are more stringent with $\text{MTTSF}_{\text{TH}} = 1000{,}000$ s and $\hat{C}_{\text{total,TH}} = 480{,}000$ hop bits/s. Table 4 lists the actual optimal values obtained as well as the optimal settings (m, T_{IDS}) under which the optimal values are obtained.

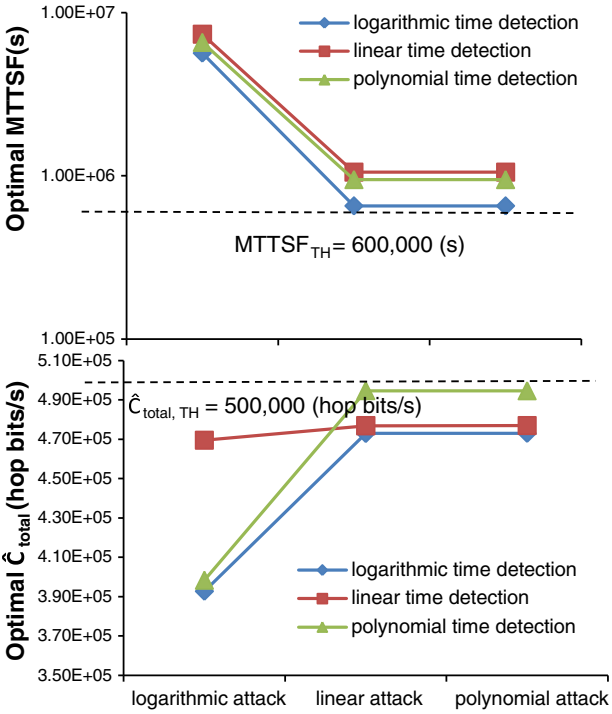


Fig. 6 Effect of detection functions on MTTSF and $\hat{C}_{\text{total}}$ under less stringent constraints

Table 3 Optimal settings for generating MTTSF and $\hat{C}_{\text{total}}$ corresponding to Fig. 6

$\text{MTTSF}_{\text{TH}} = 600{,}000$ (s) $\hat{C}_{\text{total,TH}} = 500{,}000$ (hop bits/s)	Logarithmic detection		Linear detection		Polynomial detection	
	(m, T_{IDS})	MTTSF $\hat{C}_{\text{total}}$	(m, T_{IDS})	MTTSF $\hat{C}_{\text{total}}$	(m, T_{IDS})	MTTSF $\hat{C}_{\text{total}}$
Logarithmic attack	(9,15)	5,637,970 392,698	(9, 60)	7,372,577 469,471	(9, 240)	6,565,735 398,400
Linear attack	(7, 15)	653,848 472,934	(7, 120)	1,053,356 476,857	(7, 240)	948,253 494,642
Polynomial attack	(5, 15)	653,671 473,018	(7, 120)	1,053,146 476,932	(7, 240)	948,048 494,582

In Figs. 6 and 7, we see that only linear detection is able to meet the performance constraints when the attacker function is linear or polynomial despite logarithm detection and exponential detection can generate a higher MTTSF. Consequently, the optimal setting identified is $m = 7$ and $T_{\text{IDS}} = 120$ s with the linear detection function being used to maximize MTTSF while satisfying the $\hat{C}_{\text{total,TH}}$ requirement. When the attacker function is logarithmic, we see that all three detection functions are able to satisfy the imposed performance constraints. In this case, the linear detection function with $m = 9$ and $T_{\text{IDS}} = 60$ s generates the highest MTTSF value and is selected for achieving higher survivability for the mission-oriented GCS in MANETs.

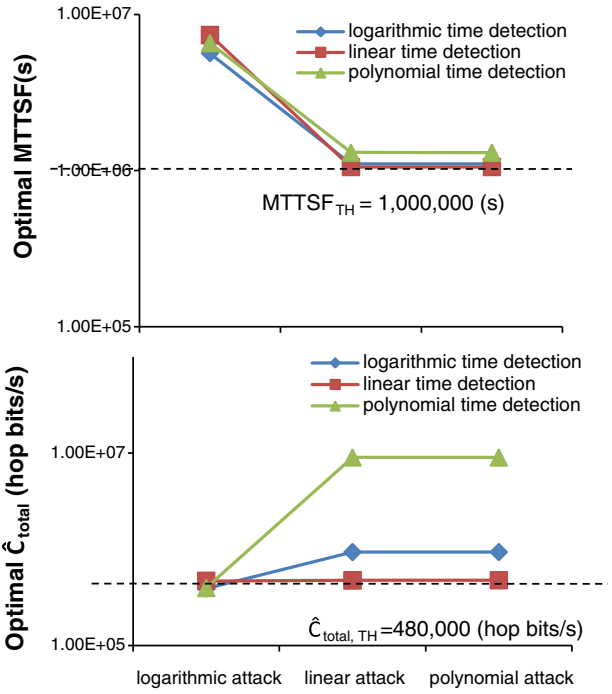


Fig. 7 Effect of detection functions on MTTSF and $\hat{C}_{\text{total}}$ under more stringent constraints

Table 4 Optimal settings for generating MTTSF and $\hat{C}_{\text{total}}$ corresponding to Fig. 7

$\text{MTTSF}_{\text{TH}} = 1{,}000{,}000$ (s) $\hat{C}_{\text{total,TH}} = 480{,}000$ (hop bits/s)	Logarithmic detection		Linear detection		Polynomial detection	
	(m, T_{IDS})	MTTSF $\hat{C}_{\text{total}}$	(m, T_{IDS})	MTTSF $\hat{C}_{\text{total}}$	(m, T_{IDS})	MTTSF $\hat{C}_{\text{total}}$
Logarithmic attack	(9, 15)	5,637,970 392,698	(9, 60)	7,372,577 469,471	(9, 240)	6,565,735 398,400
Linear attack	(9, 5)	1,104,114 938,511	(7, 120)	1,053,356 476,857	(9, 5)	1,304,663 9,041,128
Polynomial attack	(9, 5)	1,103,882 938,684	(7, 120)	1,053,146 476,932	(9, 5)	1,304,404 9,033,611

We also analyze the effect of node density on MTTSF and communication overhead. As expected (the results are not shown to cut clutter), as the node density increases, the MTTSF increases because the system will have more nodes and Condition C2 (leading to a security failure) is less likely to be met. On the other hand, as the node density increases, the group communication overhead increases because there will be more nodes actively participating in group communication activities. Lastly, as the node density increases, the IDS performance will improve because it is more likely to be able to find m vote-participants to perform voting to defend against collusion of compromised nodes.

5.2 Simulation

We have conducted extensive simulation to validate analytical results obtained using the same set of parameter values listed in Table 2. The simulation program is implemented based on a discrete-event simulation language called *SMPL* [24]. We populate the MANET area by randomly placing 150 nodes within the operational area with size $(500)^2\pi$ km^2 in our simulation. In this case, the initial number of member nodes, i.e., $N\lambda/(\lambda+\mu)$, is 120 approximately and they are scattered in the operational area. All nodes can be connected through multiple hops using a per-hop wireless radio range $= 200\,m$. Multiple groups may be observed in the operational area.

Each node in its lifecycle could generate six events, namely, GROUP JOIN, GROUP LEAVE, BEACON, GROUP COMMUNICATION, GROUP MERGE, GROUP PARTITION, INTRUSION DETECTION, and COMPROMISE. GROUP JOIN and GROUP LEAVE events occur with rates of λ and μ respectively. We assume the inter-arrival time is exponentially distributed. Upon the occurrence of a GROUP PARTITION or GROUP MERGE event, the group view and the associated membership changes are updated. The time a GROUP PARTITION event or a GROUP MERGE event occurs depends on the node distribution and user mobility. We check occurrences of group merge/partition events by a timer event. The GROUP COMMUNICATION and BEACON events are scheduled periodically with a fixed interval. The GROUP COMMUNICATION event occurs with rate λ_q. If a compromised node triggers GROUP COMMUNICATION event, we consider the system as having experienced a security failure due to violation of Condition C1. To expedite data collection, we also turn off the host-IDS capability, that is, not detecting if the sender is suspicious of compromised, so that whenever a compromised node involves in GROUP COMMUNICATION event the system fails. The INTRUSION DETECTION event occurs periodically at a rate given by the linear detection function in Eq. 4. When an INTRUSION DETECTION event occurs, each good node is tested with the false positive probability to see if it has been diagnosed by voting-based IDS as a bad node and each bad node is tested with the false negative probability to see if has been diagnosed by IDS as a bad node. Lastly, the COMPROMISE event occurs at a rate given by the linear attacker function in Eq. 3.

While we can consider any mobility model in the simulation, we adopt the *random waypoint mobility model* because of its popularity to model the movement of a node with the mobility rate being σ, the pause time being 0, and the speed being:

$$S(\sigma) = \frac{2r}{\text{expntl}(1/\sigma)} \tag{25}$$

where r is the radius of the area and $\text{expntl}(1/\sigma)$ returns a random number exponentially distributed with rate σ.

In modeling the COMPROMISE event, a good node is selected to-be-compromised (TBC) when we schedule a COMPROMISE event. A TBC node, like a compromised but undetected node, can freely join or leave a group. A TBC node then is compromised when the COMPROMISE event occurs, even if it already leaves the group it originally belongs to. If the TBC node is falsely identified as a bad node by IDS before the COMPROMISE event occurs, the COMPROMISE event is dropped and a new COMPROMISE event is scheduled right after the TBC node is evicted out of the system due to false positives of IDS. Upon a GROUP MERGE or GROUP PARTITION event, all previously scheduled COMPROMISE events in all groups involved in the group merge/partition events are canceled and a new COMPROMISE event is scheduled in each involved group.

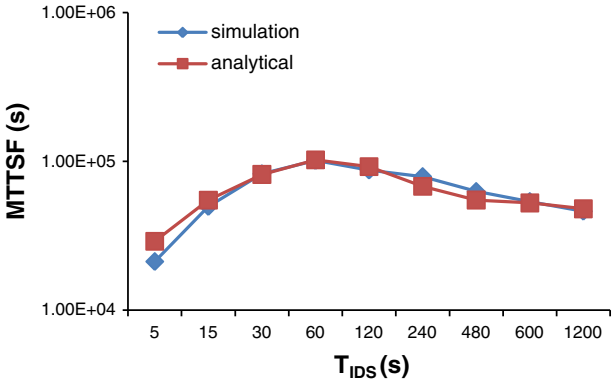


Fig. 8 Analytical versus simulation results—effect of T_{IDS} on MTTSF

A simulation run ends whenever a system failure occurs, either due to violation of Condition C1 or Condition C2. We collect the system lifetime obtained when the system failure occurs and execute another run from scratch. Figure 8 compares simulation results obtained versus analytical results for MTTSF versus T_{IDS}. The simulation results displayed are the average values out of 100 simulation runs. We see that the simulation results exhibit a similar trend compared with analytical results, identifying the optimal T_{IDS} at 60 as analytical results indicate. The mean percentage difference (MPD) between analytical results and simulation results is 10% with the standard error (SE) being 5868 (approximately 5.7% out of 102294, the optimal MTTSF at the optimal T_{IDS} identified at 60). The MPD and SE are defined by [34]:

$$MPD = \frac{\sum_{i=1}^{n} \frac{|x_i - y_i|}{y_i}}{n}, \qquad SE = \sqrt{\frac{\sum_{i=1}^{n} (x_i - y_i)^2}{n}} \tag{26}$$

where x_i is a simulation result value, y_i is an analytical result value, and n is the number of result points.

As the MPD and SE values are sufficiently small, we conclude that simulation results obtained match well with analytical results. The reason of having a slight difference between analytical and simulation results is attributed to the fact that in the analytical model we consider equal-size grouping, while in simulation groups do not necessarily have the same size. Consequently, the rate at which system failures are triggered may also be different. Nevertheless, overall we see a good correlation of simulation results versus analytical results especially in the overall trend predicted and we conclude that the analytical results obtained are valid.

6 Conclusion

In this paper, we have proposed and analyzed voting-based IDS against inside attackers for secure GCSs in MANETs. The intrusion detection interval (T_{IDS}) used by voting-based IDS can be adjusted based on the behavior of inside attackers. Our analysis revealed the intrinsic tradeoff between security (measured by the MTTSF metric) and performance (measured by the overall communication cost $\hat{C}_{\text{total}}$ metric). When given a GCS characterized by a set of parameter values, we showed that there exists an optimal detection interval (T_{IDS}) that

maximizes MTTSF as well as satisfying the constraint on the communication traffic ($\hat{C}_{\text{total}}$). The existence of the optimal detection interval (T_{IDS}) for maximizing MTTSF is attributed to the tradeoff between the positive effect of IDS (i.e., identifying compromised nodes and evicting them properly to prolong system lifetime) versus the adverse effect of IDS (i.e., false negatives and false positives generated by IDS). The existence of the optimal detection interval (T_{IDS}) for minimizing $\hat{C}_{\text{total}}$ is attributed to the tradeoff between the IDS communication traffic versus the group communication traffic.

We have investigated three ways to perform IDS detection with various vote-participants and how the system could adjust the IDS detection level in response to the attacker strength detected at runtime in order to maximize MTTSF and minimize $\hat{C}_{\text{total}}$ dynamically. By selecting the best detection function (*logarithmic*, *linear*, or *polynomial*) in response to the attacker strength, we can maximize MTTSF without experiencing much of the adverse effect of IDS. The results obtained in terms of MTTSF and $\hat{C}_{\text{total}}$ versus T_{IDS} allow the system designer to select the best intrusion detection interval (T_{IDS}) to maximize MTTSF, or minimize $\hat{C}_{\text{total}}$, depending on the security versus performance requirements, or to maximize MTTSF while satisfying the $\hat{C}_{\text{total}}$ performance requirements. To apply the results, one can cover a wide range of values of model parameters and build a table at static time listing the selection of the intrusion detection interval (T_{IDS}) that can both maximize MTTSF and/or minimize the overall communication cost ($\hat{C}_{\text{total}}$). Then, at runtime, the system can perform a table lookup operation to select the best intrusion detection function, the best IDS detection interval (T_{IDS}) and the best number of vote-participants for voting-based IDS based on statistical information collected dynamically.

In the future, we plan to enhance the adaptability of voting-based IDS to changing and evolving network environments. We are investigating the use of autonomic learning to deduce the attacker strength function based on the attacker profiles so as to apply the best detection function at the optimal setting to maximize MTTSF and/or minimize $\hat{C}_{\text{total}}$ for mobile GCSs.

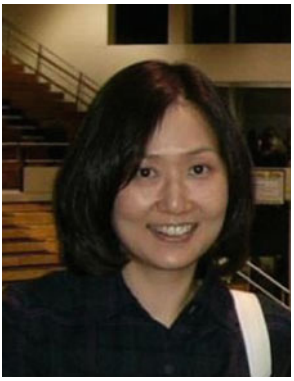
References

1. Boppana, R. V., & Su, X. (2008). An analysis of monitoring based intrusion detection for ad hoc networks. In *IEEE Global Telecommunications Conference* (pp. 1–5). New Orleans, LA.
2. Brutch, P., & Ko, C. (2003). Challenges in intrusion detection for wireless ad-hoc networks. In *Symposium on Applications and the Internet Workshops* (pp. 178–373). Orlando, FL.
3. Cabrera, J. B. D., & Gutierrez, C., & Mehra, R. K. (2005). Infrastructures and algorithms for distributed anomaly-based intrusion detection in mobile ad-hoc networks. In *IEEE Military Communications Conference* (Vol. 3, pp. 1831–1837). Atlantic City, NJ.
4. Cai, C., Guizani, S., Ci, S., & Al-Fuqaha, A. (2006). NIS02-5: Constructing an efficient mobility profile of ad-Hoc node for mobility-pattern-based anomaly detection in MANET. In *IEEE Global Telecommunications Conference* (pp. 1–5). San Francisco, LA.
5. Chan, H., Gligor, V. D., Perrig, A., & Muralidharan, G. (2005). On the distribution and revocation of cryptographic keys in sensor networks. *IEEE Transactions on Dependable and Secure Computing, 2*(3), 233–247.
6. Cho, J. H., & Chen, I. R. (2005). On design tradeoffs between security and performance in wireless group communicating systems. In *1st IEEE Workshop Secure Network Protocols* (pp. 13–18). Boston, MA.
7. Cho, J. H., & Chen, I. R. (2010). Modeling and analysis of intrusion detection integrated with batch rekeying for dynamic group communication systems in mobile ad hoc networks. *Wireless Networks*, published online.
8. Dacier, M., Deswarte, Y., & Kaâniche, M. (1996). Quantitative assessment of operational security: Models and tools. *Technical Report 96493*, Laboratory for Analysis and Architecture of Systems.
9. Debar, H., & Wespi, A. (2001). Aggregation and correlation of intrusion-detection alerts. In *4th International Symposium Recent Advances in Intrusion Detection* (pp. 85–103).

10. Gärtner, F. C. (2003). Byzantine failures and security: Arbitrary is not (always) random. *Technical Report IC/2003/20*, Swiss Federal Institute of Technology School of Computer and Communication Sciences.
11. Goseva-Popstojanova, K., Wang, F., Wang, R., Gong, F., Vaidyanathan, K. Trivedi, K., & Muthusamy, B. (2001). Characterizing intrusion tolerant systems using a state transition model. In *DARPA Information Survivability Conference and Exposition* (Vol. 2, pp. 211–221). Anaheim, CA.
12. Hasswa, A., Zulkernine, M., & Hassanein, H. (2005). Routeguard: An intrusion detection and response system for mobile ad hoc networks. *In IEEE International Conference on Wireless and Mobile Computing, Networking, and Communications* (Vol. 3, pp. 336–343)
13. Huang, Y. A., & Lee, W. (2003). A cooperative intrusion detection system for ad hoc networks. In *1st ACM Workshop on Security of Ad-hoc and Sensor Networks* (pp. 135–147). Fairfax, VA.
14. Jonsson, E., & Olovsson, T. (1997). A quantitative model of the security intrusion process based on attacker behavior. *IEEE Transactions on Software Engineering, 23*(4), 235–245.
15. Kachirski, O., & Guha, R. (2002). Intrusion detection using mobile agents in wireless ad hoc networks. In *IEEE Workshop on Knowledge Media Networking* (pp. 153–158). Kyoto, Japan.
16. Karygiannis, A., Antonakakis, E., & Apostolopoulos, A. (2006). Detecting critical nodes for MANET intrusion detection systems. In *2nd International Workshop on Security, Privacy, and Trust in Pervasive and Ubiquitous Computing* (pp. 9–15).
17. Karygiannis, T., & Owens, L. (2002). *Wireless network security: 802.11, bluetooth and handheld devices* (pp. 800–848). National Institute of Standards and Technology (NIST), Special Publication.
18. Kazienko, P., & Dorosz, P. (2004). Intrusion detection systems (IDS) part I: Network intrusions, attack symptoms, IDS tasks, and IDS architecture, http://www.windowsecurity.com/articles/intrusion_detection/.
19. Kazienko, P., & Dorosz, P. (2004). Intrusion detection systems (IDS) part II: Classification, methods, techniques, http://www.windowsecurity.com/articles/intrusion_detection/.
20. Leversage, D. J., & James, E. (2008). Estimating a system's mean time-to-compromise. *IEEE Security and Privacy, 6*(1), 52–60.
21. Li, X. S., Yang, Y. R., Gouda, M. G., & Lam, S. S. (2001). Batch rekeying for secure group communications. In *10th International Conference on World Wide Web* (pp. 525–534). Hong Kong.
22. Liu, Y., Camaniciu, C., & Man, H. (2006). A Bayesian game approach for intrusion detection in wireless ad hoc networks. In *ACM 2006 Workshop on Game Theory for Communications and Networks*, Pisa, Italy.
23. Liu, J., Yu, F. R., Lung, C. H., & Tang, H. (2009). Optimal combined intrusion detection and biometric-based continuous authentication in high security mobile ad hoc networks. *IEEE Transactions on Wireless Communications, 8*(2), 806–815.
24. MacDougall, M. H. (1987). *Simulating computer systems*. Cambridge, MA, USA: MIT Press.
25. Madan, B., Goseva-Popstojanova, K., Vaidyanathan, K., & Trivedi, K. (2002). Modeling and quantification of security attributes of software systems. In *International Conference Dependable Systems and Networks* (pp. 505–514).
26. Marti, S., Giuli, T., Lai, K., & Baker, M. (2000). Mitigating routing misbehavior in mobile ad hoc networks. In *6th Annual ACM/IEEE Mobile Computing and Networking* (pp. 255–265). Boston, MA.
27. Mishra, A., Nadkarni, K., & Patcha, A. (2004). Intrusion detection in wireless ad-hoc networks. *IEEE Wireless Communications, 11*(1), 48–60.
28. Nadeem, A., & Howarth, M. (2009). Adaptive intrusion detection and prevention of denial of service attacks in MANETs. In *International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly* (pp. 926–930). Leipzig, Germany.
29. Nicol, D. M., Sanders, W. H., & Trivedi, K. S. (2004). Model-based evaluation: From dependability to security. *IEEE Transactions on Dependable and Secure Computing, 1*(1), 48–65.
30. Perrig, A., & Tygar, J. D. (2002). *Secure broadcast communication in wired and wireless networks*. Boston: Kluwer.
31. Phoha, S. (2004). Guest editorial: Special section on mission-oriented sensor networks. *IEEE Transactions on Mobile Computing, 3*(3), 209–210.
32. Santosh, N., Saranyan, R., Senthil, K. P., & Vetriselvi, V. (2008). Cluster based co-operative game theory approach for intrusion detection in mobile ad-hoc grid. In *16th International Conference on Advanced Computing and Communications* (pp. 273–278). Chennai, India.
33. Sen S., & Clark, J. A. (2009). A grammatical evolution approach to intrusion detection on mobile ad hoc Networks. In *2nd ACM Conference on Wireless Network Security* (pp. 95–102). Zurich, Switzerland.
34. "Standard Error" definition from *Wikipedia* http://en.wikipedia.org/wiki/Standard_error_(statistics).

35. Steiner, M., Tsudik, G., & Waidner, M. (1996). Diffie-Hellman key distribution extended to group communication. In *3rd ACM Conference on Computer and Communications Security* (pp. 31–37). New Delhi, India.
36. Steiner, M., Tsudik, G., & Waidner, M. (2000). Key agreement in dynamic peer groups. *IEEE Transactions on Parallel and Distributed Systems, 11*(8), 769–980.
37. Sterne, D., Balasubramanyam, P., Carman, D., Wilson, B., Talpade, R., Ko, C., Balupari, R., Tseng, C. Y., & Bowen, T. (2005). A general cooperative intrusion detection architecture for MANETs. In *3rd IEEE International Workshop on Information Assurance* (pp. 57–70). Santa Clara, CA.
38. Subhadrabandhu, D., Sarkar, S., & Anjum, F. (2006). A framework for misuse detection in ad hoc networks. *IEEE Journal on Selected Areas in Communications, 24*(2), 274–304.
39. Sun, B., Wu, K., & Pooch, U. W. (2003). Alert aggregation in mobile ad hoc networks. In *ACM Workshop on Wireless Security* (pp. 69–78). San Diego, CA.
40. Wang, D., Bharat, D. W., Madan, B., & Trivedi, K. S. (2003). Security analysis of SITAR intrusion tolerance system. In *ACM Workshop on Survivable and Self-regenerative Systems* (pp.23–32). Fairfax, VA.
41. Yu, F. R., Tang, H., Wang, F., & Leung, V. C. M. (2009). Distributed node selection for threshold key management with intrusion detection in mobile ad hoc networks. In *International Conference on Computational Science and Engineering* (Vol. 2, pp. 787–794). Vancouver, Canada.
42. Zhang, Y., & Lee, W. (2000). Intrusion detection in wireless ad hoc networks. In *6th International Conferernce Mobile Computing and Networking* (pp. 275–283). Boston, MA.
43. Zhang, Y., Lee, W., & Huang, Y. A. (2003). Intrusion detection techniques for mobile wireless networks. *Wireless Networks, 9*(5), 545–556.

Author Biographies



Jin-Hee Cho received the B.A. from the Ewha Womans University, Seoul, Korea in 1997 and the M.S. and Ph.D. degrees in computer science from the Virginia Tech in 2004 and 2008 respectively. She is currently a postdoctoral research fellow at the US Army Research Laboratory (USARL), Adelphi Research Center, Maryland through the ARL Postdoctoral Research Program administered by the Oak Ridge Associated Universities (ORAU). 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 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, wireless systems, security, multimedia, data management, real-time intelligent systems, 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/CS and ACM.