Hierarchical Trust Management of Community of Interest Groups in Mobile Ad Hoc Networks

Ing-Ray Chen and Jia Guo
Department of Computer Science
Virginia Tech
{irchen, jiaguo}@vt.edu

***Abstract*—** In mission-critical applications deployed in mobile ad hoc networks, very frequently a commander will need to assemble and dynamically manage Community of Interest (COI) mobile groups to achieve the mission assigned despite failure, disconnection or compromise of COI members. In this paper, we present a *dynamic hierarchical trust management* protocol that can learn from past experiences and adapt to changing environment conditions (e.g., increasing misbehaving node population, evolving hostility and node density, etc.) to enhance agility and maximize application performance. With trust-based misbehaving node detection as an application, we demonstrate how our proposed COI trust management protocol is resilient to node failure, disconnection and capture events, and can help maximize application performance in terms of minimizing false negatives and positives in the presence of mobile nodes exhibiting vastly distinct QoS and social behaviors.

***Keywords*—** Hierarchical trust management; Community of interest groups; Intrusion detection.

I. INTRODUCTION

In military operation or emergency response situations, very frequently a commander will need to assemble and dynamically manage *Community of Interest* (COI) mobile groups to achieve a critical mission assigned despite failure, disconnection or compromise of COI members. COI-HM [8, 12] was proposed to achieve scalability and reconfigurability following the command chain of commander->leader->COI members. Under COI-HM, a COI is divided into multiple subtask groups to accomplish a mission. Each subtask group would be governed by a subtask group leader (SGL) dynamically appointed by the COI commander responsible for relaying commands from the commander to the COI group members in the subtask group, and filtering messages sent by COI members in the same subtask group to COI members located in other subtask groups. COI members in one subtask group may be reassigned to another subtask group for tactical reasons, thus triggering registration/deregistration actions to the subtask group leaders to maintain the hierarchical structure.

This hierarchical management structure is generic and can be applied to various mission scenarios.

Subtask groups may be physically co-located or separated. A node may be assigned to one or more subtasks, depending on node properties (e.g., manned or unmanned) and subtask group characteristics (functionality, difficulty, urgency, importance, risk, size, and composition). Thus, a node's mobility model reflects its assignment, de-assignment or reassignment to subtask groups, as well as its mobility pattern moving around the subtask groups it is assigned to. In military applications, very frequently a COI consists of heterogeneous nodes with vastly different levels of functionality, capacity and resources. A SGL is presumably a higher-capacity node and would be assigned, de-assigned, or reassigned dynamically by the COI-commander to lead a subtask group.

Despite providing scalability and reconfigurability, COI-HM does not provide tolerance against node compromises and collusion as there is no mechanism to defend against inside attackers or malicious nodes. Existing intrusion detection system (IDS) techniques based on anomaly or pattern-based detection are either centralized (especially for wired networks) which creates a single point of failure, or too complex for distributed execution in heterogeneous mobile networks at runtime.

In this paper, we propose COI *dynamic hierarchical trust management* (COI-HiTrust) for intrusion tolerance and survivability. COI-HiTrust runs on top of COI-HM, so it can achieve scalability and reconfigurability since nodes will only interact with peers in the same subtask group and do not assess trust about every node in the network. In addition as we will demonstrate later, it also achieves trust resiliency and accuracy against inside attackers or malicious nodes.

In the literature there is a large body of trust management protocols for MANETs [1, 9, 10, 16, 17, 20, 23, 27, 36-40]. However, there is very little research on hierarchically structured trust management protocol design for MANETs. Verma et al. [16] and Davis [17] considered hierarchical trust management for MANETs. However, their schemes heavily rely on the certificates issued off-line or by trusted third parties which typically are not available in MANET environments. Bao et al. [14, 24] and Li et al. [19] considered hierarchical trust management in wireless sensor networks without considering node mobility. Zhang et al. [25] proposed a hierarchical trust architecture for wireless sensor networks and considered the dynamic aspect of trust by introducing a trust varying function. However, their work is theoretical in nature without addressing what trust attributes should be used (a trust composition issue), how trust is aggregated accurately (a trust aggregation issue), or what weights should be put on trust attributes to form trust (a trust formation issue). On the contrary, our work addresses all three aspects of trust management. Moreover, we propose the concept of dynamic hierarchical trust management by which trust protocol parameter settings can be dynamically adjusted in response to changing environments to minimize trust bias and maximize application performance.

We envision the following original contributions from this work:

1. Unlike most existing reputation and trust management schemes for mobile ad hoc networks in the literature [10], we consider not only traditional "*QoS trust"* derived from communication networks, but also "*social trust"* derived from social networks [2, 3] to obtain a composite trust metric as a basis for evaluating trust of mobile nodes in COI applications. Moreover, we are the first to propose the new design notion of *mission-dependent trust formation* with the goal to enhance mission agility and maximize mission survivability despite the presence of malicious, erroneous, partly trusted, uncertain and incomplete information.
2. We design and validate COI-HiTrust as a *dynamic hierarchical trust management* protocol that can learn from past experiences and adapt to changing environment conditions (e.g., increasing or decreasing hostility, increasing misbehaving node population, etc.) to maximize application performance and enhance operation agility. This is achieved by addressing critical issues of hierarchical trust management for COI applications, namely, trust composition, aggregation, propagation, and formation. The learning process and adaptive designs of COI-HiTrust are reflected in trust aggregation, trust propagation and trust formulation. For trust composition, aggregation and propagation, we first explore novel social and QoS trust components and then devise trust aggregation and propagation protocols (for trust data collection, quality-of-information dissemination and analysis) for peer-to-peer subjective trust evaluation of *individual* social and QoS trust components, and prove the accuracy by means of theoretical analysis with simulation validation. The weights to direct trust and indirect trust are dynamically adjusted based on environment conditions to minimize trust bias. For trust formation, we explore a new design concept of *mission-dependent trust formation* allowing trust being formed out of social and QoS trust components to maximize application performance. We use a *misbehaving node detection* application as an example to illustrate the design concept. Dynamic trust management (to be discussed in more detail in Section VII) is achieved by first determining the best trust formation model given a set of model parameters specifying the environment conditions (e.g., percentage of malicious nodes), and then at runtime COI-HiTrust learns and adapts to changing environment conditions by using the best trust formation model identified from static analysis.
3. To achieve the goals of identifying the best trust composition and trust formation for mission-oriented COI mobile group applications, we develop a novel model-based analysis methodology for analyzing and validating COI-HiTrust. The novelty lies in the new design notion of *objective trust* derived from global knowledge or *ground truth* derived from the mathematical model describing a COI against

which *subjective trust* obtained as a result of executing COI-HiTrust may be compared and validated.

This paper substantially extends from [41] by adding simulation validation using ns-3 (Section V) as well as new materials, including a theoretical analysis of the protocol's convergence, accuracy and resiliency properties (Section IV), a sensitivity analysis of the effect of trust formation on application performance (Section VI), and a discussion on applicability (Section VII). The rest of the paper is organized as follows. Section II describes the system model and COI Architecture. Section III describes COI-HiTrust and explains the hierarchical trust protocol design for managing COI groups in MANETs. Section IV conducts a theoretical analysis of the convergence, accuracy and resiliency properties of COI-HiTrust. Section V develops a novel model-based approach to describe dynamic behaviors of nodes in MANETs in the presence of misbehaving nodes with the objective to yield objective trust against which subjective trust from executing COI-HiTrust may be compared for trust bias minimization. A performance evaluation is performed to demonstrate trust resiliency, convergence and accuracy properties with ns-3 simulation validation. In Section VI, we apply the hierarchical trust management protocol to *trust-based intrusion detection* to illustrate the design concept of *application performance optimization* with results and physical interpretations given. Section VII discusses how the results obtained can be applied for dynamic hierarchical trust management. Finally in Section VIII we conclude the paper and outline some future research areas.

II. System Model

2.1 COI Architecture

We assume that COI members move from one subtask to another due to task assignment, de-assignment, or reassignment, as dictated by the needs which arise during mission execution. A node can move around multiple subtask groups if it is assigned to multiple subtask groups. The node mobility model is therefore application-dependent and is given as input. A special case is that subtask groups each occupy a region in an operational area for military operation purposes. In this scenario, a mobility event occurs when a node moves across a regional boundary. One can reduce the regional size to the radio range to ensure that mobility and instability of radio environments will not be a major factor to prevent members of a subgroup from communicating directly with the SGL. Each subtask group would be governed by a SGL dynamically appointed by the COI commander. When a COI member in one subtask group moves to another subtask group, it triggers registration and deregistration actions to the two involving SGLs.

We assume that for security reasons, each node has a unique public/private key pair using Public Key Infrastructure (PKI) and the public key must remain valid throughout its lifetime as a member in the COI. PKI can effectively detect identity attacks [22] and derivatives extending from identity attacks such as Sybil

attacks, and outside attackers (i.e., attackers who are not members of the COI). However, it does not provide tolerance against compromised nodes (or inside attackers).

Our solution is COI-HiTrust (to be described in detail in Section III) with intrusion detection as an application. During mission execution, each COI member performs COI-HiTrust to evaluate trust of its peers within the same subtask group. Each SGL performs COI-HiTrust to evaluate trust of other SGLs in the COI group. A SGL collects trust evaluation results from the COI members within the subtask group and performs a summarized trust evaluation for each COI member in its subtask group. The commander collects trust evaluation results from the SGLs within the COI group and performs a summarized trust evaluation for each SGL. A SGL may be assigned, de-assigned, or reassigned depending on the evaluation result.

COI-HiTrust evaluates nodes based on both *social trust* and *QoS trust* for successful mission execution [10]. Social trust derives from the concept of *social networks* [2-4], including honesty, intimacy, selfishness, betweenness centrality, and social reputation. A COI would consist of heterogeneous mobile entities such as device-carry soldiers, robotic vehicles, or ground vehicles operated by humans. Therefore, unlike traditional network research, social trust must be considered between these mobile agents. For example, honesty is about integrity and may be considered as important as, if not more important than, competence (which is not a social trust metric) for a COI mission that concerns mission security. We use social networks to evaluate the *social trust* value of a node in terms of the degree of personal or social trends, rather than the *capability* of executing a mission based on past collaborative interactions. The latter belongs to QoS trust by which a node is judged if it is capable of completing an assigned mission as evaluated by *communication networks*. More specifically, QoS trust represents competence, dependability, reliability, successful experiences, and reputation or positive recommendations on task performance forwarded from direct or indirect interactions with others.

2.2 Attack Model

We assume that a bad node can be selfish or malicious. A malicious node is necessarily selfish. A selfish node may act uncooperatively, the degree of which depends on whom it works with (e.g., a friend or not) and whether it gains its utility. A malicious node is essentially an inside attacker who performs various attacks to disrupt the operation of a mission. A bad node can perform the following trust-related attacks to disrupt the trust system:

1. Self-promoting attacks: a malicious node can promote its importance (by providing good recommendations for itself) so as to improve its trust status. Our trust protocol deals with self-promoting attacks by considering honesty as a trust property to detect self-promoting attacks.

2. Discriminatory attacks: a socially selfish node can discriminatively attack non-friends or nodes without strong social ties because of human nature or propensity towards friends. Our trust protocol deals with discriminatory attacks by considering intimacy as a trust property.
3. Bad-mouthing attacks: a malicious node can ruin the reputation of a well-behaved node by providing bad recommendations against the good node. This is a form of collusion attacks, i.e., it can collaborate with other bad nodes to ruin the reputation of a good node. Our trust protocol deals with bad-mouthing attacks by considering honesty as a trust property.
4. Ballot-stuffing attacks: a malicious node can boost the reputation of another bad node by providing good recommendations for it. This is also a form of collusion attacks, i.e., it can collaborate with other bad nodes to boost the reputation of each other. Our trust protocol deals with ballot-stuffing attacks by considering honesty as a trust property.

A malicious node can perform Sybil and identity attacks. We assume such attacks are detected by PKI techniques [22] and the offenders will be evicted from the system. More specifically, each node has a private key and its certified public key available. A node's identity is authenticated based on the node's public/private key pair by applying the challenge/response mechanism. Consequently, every node in our COI architecture has a unique identity. A cooperative attack means that malicious nodes in the system boost their allies and focus on particular victims in the system to victimize. Bad-mouthing and ballot-stuffing attacks are a form of cooperative attacks to ruin the reputation of (and thus to victimize) good nodes and to boost the reputation of bad nodes. COI-HiTrust is said to be resilient to the above *trust-related attacks* when the "*subjective trust*" as a result of COI-HiTrust execution, is close to the "*objective trust*" despite the presence of malicious nodes performing these attacks.

III. COI-HiTrust Protocol for COI Dynamic Hierarchical Trust Management

Our protocol design starts by applying COI-HM [8, 12] by which we divide a COI into subtask groups. A node only needs to do trust evaluation of other nodes in the same subtask group. A node can send/receive messages to/from another node directly if they are in the same subtask group, or indirectly through the two nodes' SGLs if they are not in the same subtask group provided they are considered trustworthy by the SGLs, i.e., passing the runtime trust test. Similarly each SGL must pass the runtime trust test to stay in the system. Mobility management is an inherent part of COI-HM as the node mobility model represents how and when a node moves from one subtask group to another due to task assignment, de-assignment, or reassignment, as well as how and when a node moves around subtask groups if it is assigned to multiple subtask groups. For

the special case in which each subtask group occupies a region in an operational area, a mobility event occurs when a node moves across a regional boundary.

3.1 Two Levels of Subjective Trust Evaluation

COI-HiTrust maintains two peer-to-peer levels of trust in the COI-HM framework: *node-level* trust and *SGL-level* trust. Each node evaluates other nodes in the same subtask group (node-to-node) while each SGL evaluates other SGLs (SGL-to-SGL) and nodes (SGL-to-node) in its subtask group. The peer-to-peer trust evaluation is periodically updated based on either direct observations or indirect observations. When two nodes are neighbors within radio range, they evaluate each other based on direct observations via snooping or overhearing. Each node sends its trust evaluation results toward other nodes in the same subtask group to its SGL. Each SGL performs trust evaluation toward all nodes within its subtask group. Similarly, each SGL sends its trust evaluation results toward other SGLs in the COI system to the commander. The commander performs trust evaluation toward all SGLs (commander-to-SGL) in the system. A SGL is responsible for misbehaving node detection for nodes in its subtask group, while the commander is responsible for misbehaving SGL detection for all SGL nodes in the system. The commander node is one of the SGLs and is reelected periodically to remove a single point of failure. We follow the election protocol in [8] to periodically reelect the commander node from among the SGLs. The event can be triggered by other non-commander SGL nodes during the SGL-to-SGL trust evaluation process.

Our hierarchical trust management protocol dynamically performs trust evaluation and is inherently robust to mobility and instability of radio environments. Specifically, a SGL collects trust evaluation reports toward a particular member from all other members within its subtask group in each trust update interval. So a SGL will be able to perform SGL-to-node trust evaluation as long as it receives a majority of all reports. In case it does not receive enough reports in a trust interval, it can always do so in the next trust interval as soon as the radio communication is resumed.

3.2 Trust Aggregation and Propagation for Peer-to-Peer Trust Evaluation

We advocate that both social trust components such as connectivity, intimacy, honesty and unselfishness, and QoS trust components such as competence, reliability and delivery ratio be considered. Let X denote a trust component selected and let $T_{ij}^{X}(t)$ denote node i's assessment toward node j in trust property X at time t. Below we describe how trust aggregation and trust propagation for peer-to-peer trust evaluation are conducted between two COI members in the same subtask group (i.e., node-to-node trust evaluation) or two SGLs in a COI (i.e., SGL-to-SGL trust evaluation).

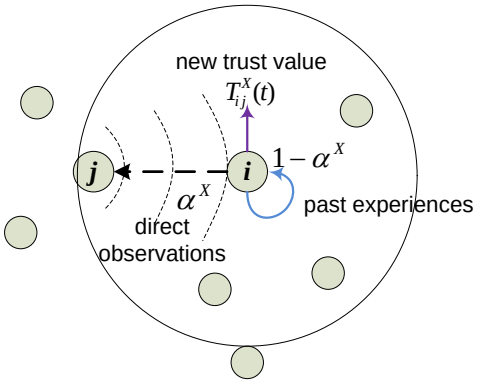

Figure 1(a): Node *i* evaluates node *j* with direct observations and past experiences in trust property *X*.

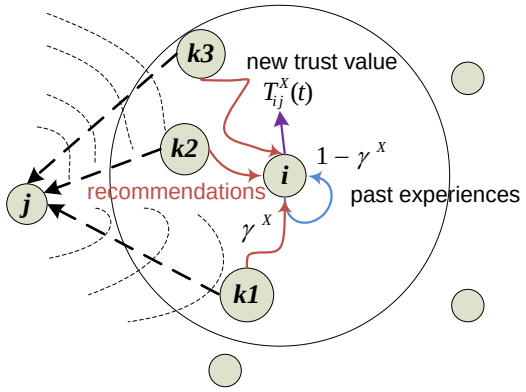

Figure 1(b): Node *i* evaluates node *j* with recommendations and past experiences in trust property *X*.

As illustrated in Figure 1, when a trustor node (node *i*) evaluates a trustee node (node *j*) in the same level at time *t*, it updates $T_{ij}^{X}(t)$ as follows:

$$T_{ij}^{X}(t) = \begin{cases} (1-\alpha^X)T_{ij}^{X}(t-\Delta t) + \alpha^X T_{ij}^{X,direct}(t) & \text{if } i \text{ and } j \text{ are 1- hop neighbors} \\ \underset{k\in N_i}{\text{avg}}\{(1-\gamma^X)T_{ij}^{X}(t-\Delta t) + \gamma^X T_{kj}^{X,recom}(t)\} & \text{otherwise} \end{cases} \quad (1)$$

In Equation 1 if node *i* is a 1-hop neighbor of node *j* at time *t*, node *i* will use its direct observations ($T_{ij}^{X,direct}(t)$) and past experiences ($T_{ij}^{X}(t-\Delta t)$ where Δt is a trust update interval) toward node *j* to update $T_{ij}^{X}(t)$. This is illustrated in Figure 1(a). We use a design parameter α^X with $0 \le \alpha^X \le 1$ to weight these two contributions and to consider trust decay over time for trust property *X*. A larger α^X means that trust evaluation will rely more on direct observations. Here $T_{ij}^{X,direct}(t)$ indicates node *i*'s trust value toward node *j* based on direct observations accumulated over the time period $[0, t]$ possibly with a higher priority given to more recent interaction experiences.

On the other hand, if node *i* is not a 1-hop neighbor of node *j*, node *i* will use its past experiences ($T_{ij}^{X}(t-\Delta t)$) and recommendations ($T_{kj}^{X,recom}(t)'s$ where *k* is a recommender) to update$T_{ij}^{X}(t)$. This is illustrated in Figure 1(b). Here $T_{kj}^{X,recom}(t)$ is the recommendation from node *k* toward node *j* in component *X* and can be just $T_{kj}^{X}(t)$. A parameter γ^X is used here to weigh these two contributions and to consider trust decay over time as follows:

$$\gamma^X = \frac{\beta^X T_{ik}(t)}{1+\beta^X T_{ik}(t)} \quad (2)$$

For ease of disposition, here we introduce another parameter $\beta^X \geq 0$ to specify the impact of "indirect recommendations" on $T_{ij}^X(t)$ such that the weight assigned to indirect recommendations is normalized to $\beta^X T_{ik}(t)$ relative to 1 assigned to past experiences. Essentially, the contribution of recommended trust increases proportionally as either $T_{ik}(t)$ or β^X increases. Instead of having a fixed weight ratio $T_{ik}(t)$ to 1 for the special case in which $\beta^X = 1$, we allow the weight ratio to be adjusted by adjusting the value of β^X and test its effect on protocol resiliency against good-mouthing and bad-mouthing attacks. Here, $T_{ik}(t)$is node *i*'s trust toward node *k* as a recommender (for node *i* to judge if node *k* provides correct information). Furthermore, to enhance QoI trust propagation, node *i* will only use its 1-hop neighbors (N_i) who are considered trustworthy (i.e., passing the RTT trust threshold) as recommenders. The new trust value $T_{ij}^X(t)$ in this case would be the average of the combined trust values of past trust information and recommendations collected at time *t*.

Our assertion is that, because different trust properties have their own intrinsic trust nature and react differently to trust decay with time, each trust property *X* has its own best (α^X, β^X) set under which subjective assessment of $T_{ij}^X(t)$ from Equation 1 would be the most accurate against actual status of node *j* in trust property *X*. To discover the best (α^X, β^X) set for trust property *X*, we resort to the development of a mathematical model describing the dynamic behavior to yield actual status of node *j*.

3.3 Mechanisms for Evaluating Direct Trust $T_{ij}^{X,direct}(t)$

In Equation 1 there is a direct trust term $T_{ij}^{X,direct}(t)$ computed by node *i* toward node *j* based on evidences observed by node *i*. For each trust property *X*, this work will develop and validate evidence-based *trust aggregation protocol*s executed by node *i* such that $T_{ij}^{X,direct}(t)$ thus obtained is accurate against actual status of node *j* at time *t*. Below we describe trust aggregation protocols by which node *i* can collect evidences to assess $T_{i,j}^{X,direct}(t)$ for the case in which *i* and *j* are 1-hop neighbors at time *t* for *X*=intimacy, honesty, unselfishness (social components) and competence (a QoS component) below.

- $T_{i,j}^{intimacy,direct}(t)$: This measures intimacy or closeness of node *i* toward node *j*. It follows the maturity model proposed in [1] in that the more interaction experiences *A* had with *B*, the more trust and confidence *A* will have toward *B*. The mechanism for node *i* to compute $T_{i,j}^{intimacy,direct}(t)$ is as follows: If node *j* is a friend to node *i* as deriving from a "friendship" matrix [6], then $T_{i,j}^{intimacy,direct}(t) = 1$. Otherwise node *i* computes $T_{i,j}^{intimacy,direct}(t)$ by the ratio of the number of interactions it has with node *j* during $[t - d\Delta t, t]$ to the maximum number of interactions with any other node. Here *d* is the window size giving

recent interaction experiences higher priority over ancient experiences. Note that for encounter-based COI applications, encountering experiences are interaction experiences. In this case, $T_{i,j}^{intimacy,direct}(t)$ is computed by the ratio of the amount of time nodes *i* and *j* are 1-hop neighbors during $[t - d\Delta t, t]$.

- $T_{i,j}^{honesty,direct}(t)$: This refers to the belief of node *i* that node *j* is honest based on node *i*'s direct observations during $[t - d\Delta t, t]$. The mechanism for node *i* to estimate $T_{i,j}^{honesty,direct}(t)$ is beta distribution as commonly used in Bayesian Inference [26]. Specifically, node *i* uses a set of anomaly detection rules such as interval, retransmission, jamming and delay rules [13, 21] to keep a count of compliant experiences of node *j* (called A) and a count of suspicious experiences of node *j* (called B) during $[t - d\Delta t, t]$. Node *i* then computes $T_{i,j}^{honesty,direct}(t)$ by A/(A+B). If A+B=0, meaning there are neither compliant nor suspicious experiences observed over $[t - d\Delta t, t]$, then $T_{i,j}^{honesty,direct}(t)$ is set to 0.5 meaning no new knowledge is available. This will result in trust decay over time based on Equation 1.
- $T_{i,j}^{unselfishness,direct}(t)$: This provides the belief of node *i* that node *j* is unselfishness based on direct observations during $[t - d\Delta t, t]$. The mechanism for node *i* to estimate $T_{ij}^{unselfishness,direct}(t)$ is also Bayesian Inference [26]. Specifically, node *i* estimates $T_{ij}^{unselfishness,direct}(t)$ by the ratio of the number of cooperative interaction experiences to the total number of protocol interaction experiences. Note that both counts are related to protocol execution except that the former count is for positive experiences when node *j*, as observed by node *i*, cooperatively follows the prescribed protocol execution.
- $T_{i,j}^{competence,direct}(t)$: This refers to the belief of node *i* that node *j* is competent at time *t*. The mechanism for node *i* to estimates $T_{i,j}^{competence,direct}(t)$ is also Bayesian Inference [26]. Specifically, node *i* overhears node *j*'s packet transmission activities during $[t - d\Delta t, t]$ and measures the transmission delay experienced each time. If the delay measured is within the normal range, it is recorded as a positive experience in competence; otherwise it is a negative experience. Node *i* then estimates $T_{ij}^{competence,direct}(t)$ by the ratio of the number of positive packet transmission experiences to the total number of packet transmission experiences. In practice, as long as node *j* is alive (energy is not depleted and there is no hardware/software failure), node *j* is competent.

The above trust aggregation protocols will be tested for their validity. An important task is to assess the accuracy of $T_{ij}^{X,direct}(t)$ obtained. We compare $T_{ij}^{X,direct}(t)$ with $T_j^X(t)$, i.e., the actual status of node *j* at time *t* in trust property *X*. The latter quantity can be obtained by defining a continuous-time semi-Markov

process describing the dynamic behavior of node *j* and thus yielding actual status of node *j* at time *t*. This provides a basis for validating trust aggregation designs such that $T_{ij}^{X,direct}(t)$ computed is as close to actual status of *j* as possible. The difference between $T_{ij}^{X,direct}(t)$ and $T_j^X(t)$ is the *direct trust assessment error*, $TE_{ij}^{X,direct}(t)$, defined as follows:

$$TE_{ij}^{X,direct}(t) = T_{ij}^{X,direct}(t) - T_j^X(t) \quad (3)$$

$TE_{ij}^{X,direct}(t)$ above is one source of trust inaccuracy. Another source of trust inaccuracy is due to compromised nodes providing incorrect trust recommendations through bad-mouthing and ballot-stuffing attacks. The difference between $T_{ij}^X(t)$ (from Equation 1) and $T_j^X(t)$ is the *trust bias* in component *X*, $TB_{ij}^X(t)$, defined as follows:

$$TB_{ij}^X(t) = T_{ij}^X(t) - T_j^X(t) \quad (4)$$

$TB_{ij}^X(t)$ is the end result of a trust aggregation protocol execution. The goal of trust propagation protocol design is to minimize $TB_{ij}^X(t)$. We minimize $TB_{ij}^X(t)$ by dynamically selecting the best set of (α^X, β^X) set under which subjective assessment of $T_{ij}^X(t)$ from Equation 1 would be the most accurate against actual status of node *j* in trust property *X*, i.e., $TB_{ij}^X(t)$ is minimized.

3.4 Trust Formation

We advocate *trustee-dependent trust formation* by which the best way to form trust from social trust and QoS trust is identified and applied to each individual node, properly reflecting trustee properties given as input. We also advocate *mission-dependent trust formation* by which the best way to form trust from social trust and QoS trust is identified and applied to each individual subtask group to maximize application performance, properly reflecting subtask group mission characteristics given as input.

Let $T_{ij}(t)$ denote node *i*'s trust toward node *j* at time *t*. To form trust from social trust and QoS trust, let $T_{ij}^{social}(t)$ and $T_{ij}^{QoS}(t)$ denote node *i*'s social trust and QoS trust toward node *j* at time *t*, respectively, derived from $T_{ij}^X(t)$ in Equation 1. We will explore the *importance-weighted-sum* trust formation with which trust is an importance-weighted sum of social trust and QoS trust. It encompasses more-social-trust, more-QoS-trust, social-trust-only, and QoS-trust-only in trust formation. It is particularly applicable to missions where context information is available about the importance of social or QoS trust properties for successful mission execution. For example, for a subtask group consisting of unmanned nodes, the more-QoS-trust or QoS-trust-only trust formation model will be appropriate. Specifically,

$$T_{ij}(t) = w^{social} T_{ij}^{social}(t) + w^{QoS} T_{ij}^{QoS}(t) \quad (5)$$

where w^{social} and w^{QoS} are "importance" weights associated with social trust and QoS trust, respectively, with $w^{social} + w^{QoS} = 1$.

Note that in the above formulation, $T_{ij}^{social}(t)$ and $T_{ij}^{QoS}(t)$ are *aggregate trust* derived from Equation 1 with *X*=social trust components and *X*=QoS trust components, respectively. We explore the weighted-sum-form model to aggregate $T_{ij}^{social}(t)$ and $T_{ij}^{QoS}(t)$ to allow the relative importance of each trust property in its category (social or QoS) to be specified. As an example, suppose that a mission dictates intimacy and honesty be picked as two social trust properties and both are considered equally important. Then, $T_{ij}^{social}(t)$ would be computed by $T_{ij}^{social}(t) = 0.5\, T_{ij}^{intimacy}(t) + 0.5\, T_{i,j}^{honesty}(t)$.

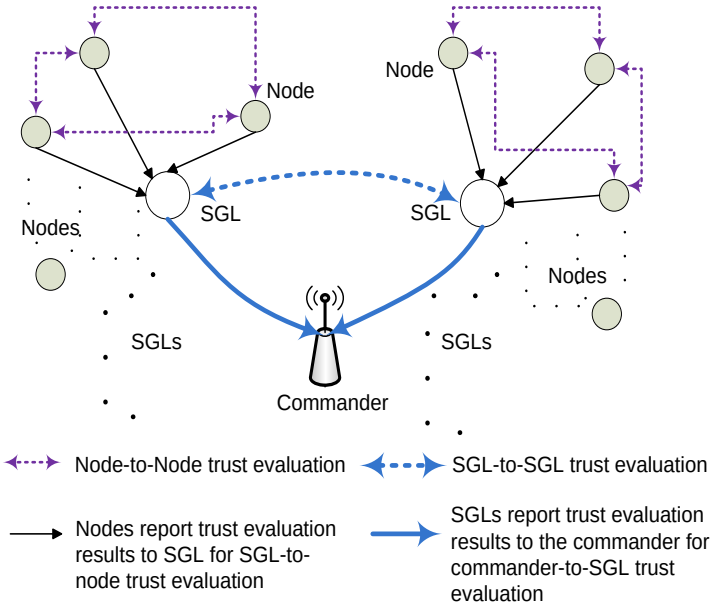

Figure 2: Information Flow of Hierarchical Trust Evaluation in COI-HiTrust.

3.5 Hierarchical Trust Evaluation

Figure 2 illustrates the information flow of hierarchical trust evaluation in COI-HiTrust with node-to-node, SGL-to-node, SGL-to-SGL and commander-to-SGL trust evaluation. Leveraging the COI-HM structure, each node reports its trust evaluation toward other nodes in the same subtask group to its SGL, possibly through trust-based routing to counter black-hole attacks. The SGL then applies statistical analysis principles to $T_{ij}(t)$ values received to perform SGL-to-node trust evaluation towards node j to yield $T_j^{COI-HiTrust}(t)$. One application-level trust setting design is to set a drop-dead minimum trust threshold

T^{th}. A SGL, say node *c*, takes $T_{ij}(t)$ from node *i* only if it considers node *i* is trustworthy, i.e., $T_{ci}(t)>T^{th}$. Then it can compute $T_j^{COI-HiTrust}(t)$ for node *j* as the average of $T_{ij}(t)'s$ from trustworthy nodes in its subtask group, i.e., be appropriate. Specifically,

$$T_j^{COI-HiTrust}(t) = \underset{i\in N_R \wedge T_{ci}(t)\geq T^{th}}{avg}\{T_{ij}(t)\} \quad (6)$$

where N_R is the set of COI members in the subtask group. The SGL (node *c*) if certified to be trustworthy by the commander could announce *j* as compromised if $T_j^{COI-HiTrust}(t)$ is less than T^{th}; otherwise, node *j* is not compromised. This is discussed more in Section IV *Misbehaving Node Detection* below. Also the SGL may leverage $T_{ij}(t)$ values received to detect if there is any outlier as an evidence of good-mouthing or bad-mouthing attacks.

IV. THEORETICAL ANALYSIS

In this section we formally prove the convergence, accuracy, and resiliency properties of our trust management protocol against trust attacks. We first provide proofs for the case in which the environment is stationary, i.e., the node status (good vs. malicious) is stationary. Then we extend the proof to the case in which the environment is non-stationary. Later in Section V, we present ns-3 simulation results to validate the convergence, accuracy, and resiliency properties of our protocol design in non-stationary environments.

For ease of disposition, we simplify the notations $T_{ij}^{X,direct}(t)$ to $D_{ij}^X(t)$ (with *D* standing for direct trust), and $T_{kj}^{X,recom}(t)$ to $R_{kj}^X(t)$ (with *R* standing for recommendation trust). Also for ease of disposition, we omit the superscript X in $T_{ij}^X(t)$, $D_{ij}^X(t)$, and $R_{kj}^X(t)$, since the analysis is generic and applicable to all trust properties.

We consider the *instantaneous trust* (reflecting the actual behavior) of a node *j* as a stochastic process $G_j = \{G_j(n): n = 1,2,\dots\}$ where *n* is the *nth* trust update interval, i.e., $t = \Delta t_1 + \Delta t_2 + \cdots + \Delta t_n$. We define *objective trust* (or ground truth trust) at step *n* as the expected value of $G_j(n)$, i.e., $E[G_j(n)]$. The goal of trust management is to estimate *objective trust* using direct observations on a node's instantaneous behaviors and recommendations. The direct trust of node *i* towards node *j* (also a stochastic process $D_{ij} = \{D_{ij}(n): n = 1,2,\dots\}$ might be different from G_j due to noises. We assume the noise process ε_{ij} is a white noise with $\varepsilon_{ij}(n): n = 1,2,\dots$ being independent and identically distributed and drawn from a zero-mean distribution. Then,

$$D_{ij}(n) = G_j(n) + \varepsilon_{ij}(n) \quad (7)$$

Assume that G_j and D_{ij} are random variables in the space of [0, 1], so ε_{ij} is in the space of [-1, 1]. That is, if G_j=1 but $D_{ij} = 0$, then $\varepsilon_{ij} = -1$. if G_j=0 but $D_{ij} = 1$, then $\varepsilon_{ij} = 1$. Our proof does not rely on the specific distributions of these random processes. For example, for the *honesty* trust property, a node might behave honest or dishonest, i.e., G_j follows Bernoulli distribution; for the *cooperativeness* trust property, G_j could be a random variable in the space of [0, 1] following a Beta distribution.

According to our trust aggregation and propagation protocol described in Section 3.2, the subjective trust evaluation of node i towards node ($T_{ij} = \{T_{ij}(n): n = 1,2, \dots\}$) is updated by either direct observations D_{ij} or recommendations from another node k ($R_{kj} = \{R_{kj}(n): n = 1,2, \dots\}$).

When node i directly interacts with node j, we have:

$$T_{ij}(n) = (1 - \alpha)T_{ij}(n-1) + \alpha D_{ij}(n) \quad (8)$$

Otherwise, if node i interacts with another node k, we have:

$$T_{ij}(n) = (1 - \gamma)T_{ij}(n-1) + \gamma R_{kj}(n) \quad (9)$$

where $\gamma = \frac{\beta D_{ik}(n)}{1+\beta D_{ik}(n)}$.We analyze the trust accuracy, convergence and resiliency properties of our protocol based on trust bias, i.e., $b_{ij}(n) = T_{ij}(n) - G_j(n)$. We define trust accuracy, convergence and resiliency by $\lim_{n\to\infty} E\left[b_{ij}(n)\right]$, i.e., how much the subjective trust deviates from the objective trust when it converges in the presence of malicious attacks. Note that as $\lim_{n\to\infty} E\left[b_{ij}(n)\right]$ converges, the variance will be stabilized, leading to an upper bound of trust bias that defines trust accuracy achieved.

We first consider a stationary environment in which G_j and D_{ij} are stationary random processes. We also first consider a simple case in which there is no trust-related attack, i.e., $R_{kj}(n) = D_{kj}(n)$ or the recommended trust of node *k* toward node *j* is equal to the direct trust of node *k* toward node *j*. Later on we will relax these assumptions.

If node i interacts with a node at stage n, there are two ways to update trust:

(1) If node i directly interacts with node j, it uses direct observations to update its trust toward node j according to Equation 1. So the trust bias of node i towards node j at stage n is:

$$b_{ij}(n) = T_{ij}(n) - G_j(n) \ = (1 - \alpha)T_{ij}(n-1) + \alpha D_{ij}(n) - G_j(n) \quad (10)$$

Since $D_{ij}(n) = G_j(n) + \varepsilon_{ij}(n)$, we have

$$b_{ij}(n) = (1-\alpha)T_{ij}(n-1) + \alpha\left(G_{ij}(n) + \varepsilon_{ij}(n)\right) - G_j(n) \quad (11)$$

Reformatting the equation above by subtracting and adding $(1-\alpha)G_j(n-1)$ in the right side, we have:

$$b_{ij}(n) = (1-\alpha)\left(T_{ij}(n-1) - G_j(n-1)\right) + (1-\alpha)G_j(n-1) + \alpha\left(G_j(n) + \varepsilon_{ij}(n)\right) - G_j(n) \quad (12)$$

Since $b_{ij}(n-1) = T_{ij}(n-1) - G_j(n-1)$, we have:

$$b_{ij}(n) = (1-\alpha)b_{ij}(n-1) + (1-\alpha)G_j(n-1) - (1-\alpha)G_j(n) + \alpha\varepsilon_{ij}(n) \quad (13)$$

(2) If node i interacts with another node k, rather than j at stage n, it uses the recommendation from node k to update its trust toward node j according to Equation 1. Using a similar derivation as above, we have the trust bias of node i towards node j at stage n as follows:

$$b_{ij}(n) = (1-\gamma)b_{ij}(n-1) + (1-\gamma)G_j(n-1) - (1-\gamma)G_j(n) + \gamma\varepsilon_{kj}(n) \quad (14)$$

Note that in this derivation above, we assume there is no trust-related attack, so $R_{kj}(n) = D_{kj}(n) = G_j(n) + \varepsilon_{kj}(n)$.

Since we assume a stationary environment and a zero mean noise, we have $E\left[G_j(n-1) - G_j(n)\right] = 0$, $E\left[\varepsilon_{ij}(n)\right] = 0$, and $E\left[\varepsilon_{kj}(n)\right] = 0$. Let p ($0 \le p \le 1$) denote the probability that node i interacts with node j at stage n. Then, with the independence assumption, we have:

$$E\left[b_{ij}(n)\right] = \{p(1-\alpha) + (1-p)(1-E[\gamma])\}E\left[b_{ij}(n-1)\right] = \Theta E\left[b_{ij}(n-1)\right] \quad (15)$$

Since $D_{ik}(n) = G_k(n) + \varepsilon_{ik}(n)$ is independent of $b_{ij}(n-1)$, $\gamma = \frac{\beta D_{ik}(n)}{1+\beta D_{ik}(n)}$ is independent of $b_{ij}(n-1)$. Consequently, as long as $0 \le \Theta < 1$, $\lim_{n\to\infty} E\left[e_{ij}(n)\right] = 0$, i.e., the trust evaluation converges. To make sure $0 \le \Theta < 1$, we need $0 < \alpha \le 1$ and $0 < E[\gamma] \le 1$. Notice that $0 < E[\gamma] = E\left[\frac{\beta D_{ik}(n)}{1+\beta D_{ik}(n)}\right] \le 1$ if $\beta > 0$. Hence, we have Lemma 1 as follows:

Lemma 1: *In a stationary environment, if there is no trust-related attacks, the trust evaluation in our trust management protocol converges as long as* $0 < \alpha \le 1$ and $\beta > 0$.

From Equation 15, we note that the trust evaluation converges exponentially when $0 \le \Theta < 1$. Thus, the convergence speed increases as Θ decreases. This leads to Lemma 2.

Lemma 2: *In a stationary environment, if there is no trust-related attacks, the trust convergence speed of our trust management protocol increases as* α *or* β *increases* ($0 < \alpha \leq 1$ and $\beta > 0$).

We measure trust fluctuation by the variance of trust bias, i.e., $Var\left[T_{ij}(n) - E\left[G_j(n)\right]\right]$. Consider a stationary environment, we have $Var\left[T_{ij}(n) - E\left[G_j(n)\right]\right] = Var\left[T_{ij}(n)\right]$. Below we analyze the effects of trust parameters on trust fluctuation. Again, we consider two cases based on the node with which node i interacts:

(1) If node i interacts with node j, it uses direct observations to update trust according to Equation 1. Then, we have,

$$T_{ij}(n) = (1-\alpha)T_{ij}(n-1) + \alpha\left(G_j(n) + \varepsilon_{ij}(n)\right) \tag{16}$$

$T_{ij}(n-1)$ is obtained based on past direct observations and trust bias at step $n-1$, so it is independent of $G_j(n)$ and $\varepsilon_{ij}(n)$, which are also independent of each other. Therefore:

$$Var\left[T_{ij}(n)\right] = (1-\alpha)^2 Var\left[T_{ij}(n-1)\right] + \alpha^2\left(Var\left[G_j(n)\right] + Var\left[\varepsilon_{ij}(n)\right]\right) \tag{17}$$

Since $G_j(n)$ and $\varepsilon_{ij}(n)$ are stationary random processes, $Var\left[G_j(n)\right] + Var\left[\varepsilon_{ij}(n)\right]$ is constant. Therefore, after trust convergence ($0 < \alpha \leq 1$ and $n \to \infty$), the variance of trust evaluation will stabilize to a constant value, i.e., $Var\left[T_{ij}(n)\right] = Var\left[T_{ij}(n-1)\right]$, that is,

$$Var\left[T_{ij}(n)\right] = (1-\alpha)^2 Var\left[T_{ij}(n)\right] + \alpha^2\left(Var\left[G_j(n)\right] + Var\left[\varepsilon_{ij}(n)\right]\right) \tag{18}$$

This simplifies to:

$$Var\left[T_{ij}(n)\right] = \frac{\alpha}{2-\alpha}\left(Var\left[G_j(n)\right] + Var\left[\varepsilon_{ij}(n)\right]\right) \tag{19}$$

From Equation 19, we can see that in this case, after trust convergence, the trust fluctuation increases as α increases.

(2) If node i interacts with another node k, rather than j at stage n, it uses the recommendation from node k to update trust in accordance with Equation 1. Using a similar derivation as in case (1), we have

$$Var\left[T_{ij}(n)\right] = \frac{\gamma}{2-\gamma}\left(Var\left[G_j(n)\right] + Var\left[\varepsilon_{ij}(n)\right]\right) \tag{20}$$

Since $\gamma = \frac{\beta D_{ik}(n)}{1+\beta D_{ik}(n)}$ in this case, after trust convergence, the trust fluctuation increases as β increases.

Because the trust update falls into either case (1) or case (2) above, we have Lemma 3 as follows:

Lemma 3: *In a stationary environment, if there is no trust-related attacks, the variance or trust fluctuation of the trust value after convergence in our trust management protocol increases as* α *or* β *increases* ($0 < \alpha \leq 1, \beta > 0$).

Now we extend the analysis to the more general case when there are malicious nodes performing trust related attacks. Because of attacks, the trust evaluation may not converge to objective trust. However, we can select appropriate trust parameters such that the trust evaluation converges and the trust bias is minimized. Suppose that the percentage of malicious nodes in the network is λ and the probability that node i interacts with node j at stage n is p. Again, there are two cases:

(1) If node i interacts with node j, it uses direct observations to update trust. Then following the same derivation for Equation 13, the trust bias of node i towards node j at stage n is:

$$b_{ij}(n) = (1-\alpha)b_{ij}(n-1) + (1-\alpha)G_j(n-1) - (1-\alpha)G_j(n) + \alpha\varepsilon_{ij}(n) \qquad (21)$$

(2) If node i interacts with another node k (who had prior interaction experience with node j) rather than with node j itself at stage n, it uses the recommendation from node k to update trust. Following the same derivation for Equation 14, we have the trust bias of node i towards node j at stage n as follows:

$$b_{ij}(n) = (1-\gamma)b_{ij}(n-1) + (1-\gamma)G_j(n-1) - (1-\gamma)G_j(n) + \gamma\varepsilon'_{kj}(n) \qquad (22)$$

Equation 22 is similar to Equation 14 except that we have used $\varepsilon'_{kj}(n)$ instead of $\varepsilon_{kj}(n)$ to account for possible recommendation attacks from node k. Let λ be the percentage of malicious nodes. With probability 1- λ, node k is a good node in which case $E\left[\varepsilon'_{kj}(n)\right] = E\left[\varepsilon_{ij}(n)\right] = 0$ as before. With probability λ, node k is a malicious node in which case there is a non-zero-mean trust-related recommendation attack noise and $E\left[\varepsilon'_{kj}(n)\right] \neq 0$. Meanwhile we still have $E\left[G_j(n-1) - G_j(n)\right] = 0$ since we assume a stationary environment. Summarizing above, the expected value of $b_{ij}(n)$ in Equation 22 is given by:

$$E\left[b_{ij}(n)\right] = \{p(1-\alpha) + (1-p)(1- E[\gamma])\}\, E\left[b_{ij}(n-1)\right] + \lambda(1-p)E[\gamma]E[\varepsilon'_{kj}(n)] \qquad (23)$$

We can see from Equation 23 that as long as $0 \leq \Theta = p(1-\alpha) + (1-p)(1-E[\gamma]) < 1$, $E\left[b_{ij}(n)\right]$ will eventually converge. Therefore, $E\left[b_{ij}(n)\right] = E\left[b_{ij}(n-1)\right]$ eventually at which point we will have:

$$E\left[b_{ij}(n)\right] = \frac{\lambda(1-p)E[\gamma]}{1-\left(p(1-\alpha)+(1-p)(1-E[\gamma])\right)} E[\varepsilon'_{kj}(n)] \quad (24)$$

Reformatting the equation above, we have:

$$\left|E\left[b_{ij}(n)\right]\right| = \frac{\lambda(1-p)E[\gamma]}{\alpha p+(1-p)E[\gamma]} \left|E[\varepsilon'_{kj}(n)]\right| < \lambda\left|E[\varepsilon'_{kj}(n)]\right| \quad (25)$$

Here $E[\gamma] = E\left[\frac{\beta D_{ik}(n)}{1+\beta D_{ik}(n)}\right]$ <1 and $-1 < E[\varepsilon'_{kj}(n)] < 1$ since in our protocol, a trust value is in the range of [0, 1]. Therefore, $\left|E\left[b_{ij}(n)\right]\right| < \lambda$. Equation 25 leads to Lemma 4 as follows:

Lemma 4: *In a stationary environment, if there are malicious nodes performing trust related attacks, the trust evaluation in our trust management protocol stabilizes as long as* $0 < \alpha \leq 1$ and $\beta > 0$. *The trust bias is less that* λ *after trust stabilizes and decreases as* α *increases or as* β *decreases.*

Now we extend the proof to non-stationary environments in which G_j and D_{ij} are non-stationary random processes. We note that the objective trust status may change before trust converges since trust convergence and stabilization take time. However, from Equations 15 and 23, subjective trust obtained as a result of executing our trust management protocol will approach objective trust even if objective trust changes dynamically, and will converge to objective trust if objective trust stabilizes after each change. Hence, as long as we select high α and β to shorten trust convergence time at the expense of high trust fluctuation, the system operating under our protocol will quickly adapt to environment changes. □

V. TRUST PROTOCOL PERFORMANCE

We develop a mathematical model based on *continuous-time semi-Markov stochastic processes* each modeling a mobile node in the COI mission-oriented group. Specifically, we leverage the stochastic Petri net techniques [7, 31-35] to define a continuous-time semi-Markov process describing the status of a node as time progresses, including the subtask group the node resides, compromise status, selfishness status, hardware/software failure status, and energy status, thus providing global information regarding the probability that the node is in a particular subtask group, whether it is compromised or not, whether it is selfish or not, and whether it is still alive and thus competent to perform the mission assigned at time *t*. A node is considered incompetent when its energy is depleted or it suffers from a hardware/software failure. Each node is characterized by its specific mobility model (representing movements between subtask groups), compromise rate, selfish rate, hardware/software failure rate, initial energy (a SGL has more resources and more energy than a regular nodes), and role-based energy consumption rate. Thus, a node's

semi-Markov stochastic process must reflect the node's specific characteristics in addition to the COI's operational and environmental characteristics. Moreover, a node with its own stochastic process will go from one state to another, depending on its interactions with other nodes, each having its own continuous-time semi-Markov process. We develop an iterative computational procedure so that all semi-Markov stochastic processes converge, thus properly reflecting node interaction experiences with each other. The output of the mathematical model is the objective trust function for each trust property X for node j at time t, i.e., $T_j^{X,OBJ}(t)$ from which we obtain objective trust $T_j{}^{OBJ}(t)$ based on a trust formation model (e.g., Equation 5) to be compared with subjective trust $T_j^{COI-HiTrust}(t)$ obtained in Equation 6 for accuracy assessment.

Table 1 lists the default parameter values. We consider an environment with N = 400 heterogeneous mobile objects/devices, each with energy E drawn from uniform distribution U[12, 24] hours. Devices are connected in a social network represented by a friendship matrix [6]. Physically, nodes move according to the SWIM mobility model [5] modeling human social behaviors in 16×16 regions with the length of each region equal to radio range R, so that two nodes are neighbors is they are in the same region or in the neighbor regions. We consider the case in which a 4×4 area is a subtask group area. Each node is subject to dishonesty and selfishness behavior attacks with rates λ_{com} and $\lambda_{selfish}$ respectively. Initially All nodes are honest and unselfish, but may turn into dishonest and selfish as time progresses depending on the attack rates. A dishonest node performs self-promoting, bad-mouthing and ballot-stuffing attacks as described in Section II. A selfish node performs discriminatory attacks. All nodes runs COI-HiTrust to perform peer-to-peer trust evaluation with the update interval Δt = 0.2hr with the observation window size d=2.

In the following we report analytical results based on Equations 1-6. We further experimentally validate analytical results with extensive simulation using ns-3 [18, 28].

Table 1: Parameters and Default Values/Ranges used.

Parameter	Value	Parameter	Value
E	U[12, 24] hr	R	250 m
speed	1.45	N	400
pause	2 hrs	d	2
λ_{com}	1/18hrs	Δt	0.2 hr
$\lambda_{selfish}$	1/36hrs	$TE_{ij}^{X,direct}$	U[0, 0.2]

5.1 P2P Trust Accuracy and Convergence Behavior

We examine peer-to-peer trust convergence behavior of our trust protocol design. Figure 3 plots $TB_{ij}^{honesty}(t)$ defined by Equation 4, i.e., the difference between subjective $T_{ij}^{honesty}(t)$ (from Equation 1) and objective $T_j^{honesty}(t)$ (ground truth) over time for a trustor node (i.e., node *i*) and a trustee node (i.e., node j) randomly picked. The solid lines are analytical results and the dashed lines are ns-3 simulation results.

The subjective trust is obtained from COIHiTrust operating at the identified optimal ($\alpha^{honesty}$, $\beta^{honesty}$) settings as shown in Table 2. We observe that a very distinct set of ($\alpha^{honesty}$, $\beta^{honesty}$) is being used by the trustor node in response to the attacker strength detected at runtime. Specifically, when the attacker strength is strong, the trustor node would rather trust its own assessment and hence it uses a high $\alpha^{honesty}$(in the range of [0, 1]) and a low $\beta^{honesty}$ (in the range of [1, 10]) at the expense of slow trust convergence. Conversely, when the environment condition is benign, the trustor node would rather take in more trust recommendations and hence it uses a low $\alpha^{honesty}$ and a high $\beta^{honesty}$ so it can quickly achieve trust convergence without risking trust inaccuracy. There are several curves in Figure 3, each with a different compromise rate λ_{com} representing the attacker strength. We see that trust bias $TB_{ij}^{honesty}(t)$ is higher as the compromise rate λ_{com} increases because there are more compromised nodes in the system performing trust-related attacks to disrupt the trust system. Nevertheless, we see a stable convergence behavior of our trust protocol with trust bias limited to 0.07 even for a high compromise rate. The mean square error (MSE) between $T_{ij}^{honesty}(t)$ and $T_j^{honesty}(t)$ is small as shown in Table 2.

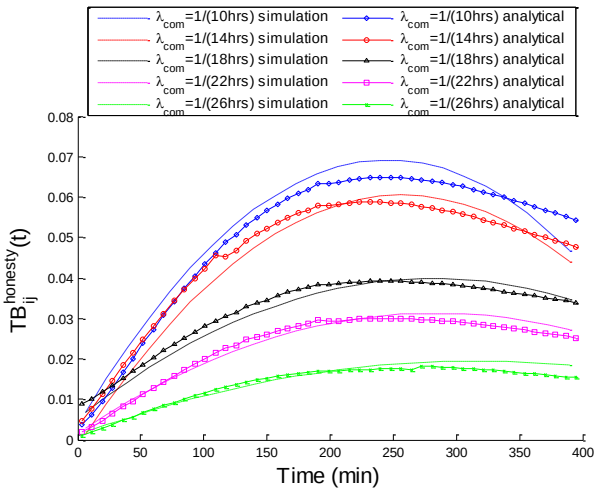

Figure 3: $TB_{ij}^{honesty}(t)$ over time.

Table 2: ($\alpha^{honesty}$, $\beta^{honesty}$) setting to minimize trust bias.

λ_{com}	($\alpha^{honesty}$, $\beta^{honesty}$)	MSE of trust bias
1/(10hrs)	(0.9, 1)	0.0123
1/(14hrs)	(0.9, 1)	0.0131
1/(18hrs)	(0.7, 2)	0.0083
1/(22hrs)	(0.6, 4)	0.0076
1/(26hrs)	(0.65, 5)	0.00053

We observe a remarkable match between the analytical results (solid lines) and the simulation results (dashed lines). The close match validates our analytical model and verifies the validity of our trust protocol design against trust-related attacks (self-promoting, bad-mouthing, and ballot-stuffing attacks) by malicious nodes.

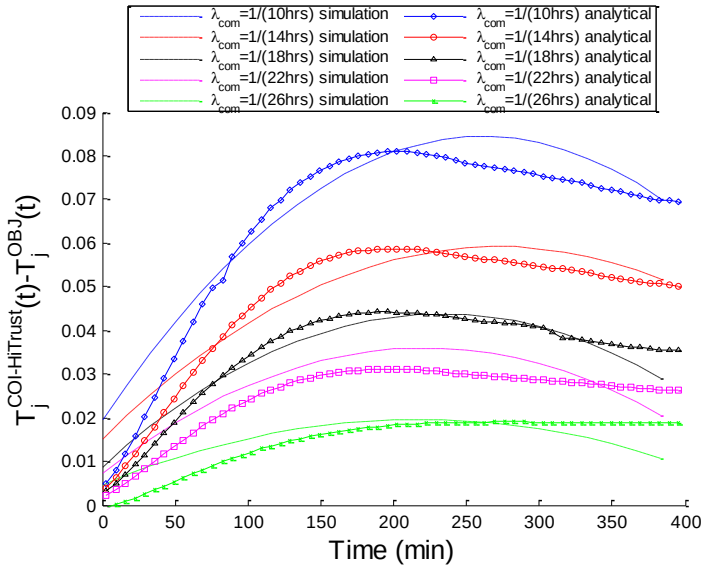

Figure 4: Trust bias of COIHiTrust over time.

5.2 COI-HiTrust Accuracy and Convergence Behavior

In this section, we examine the trust convergence and accuracy behavior of COIHiTrust by ns-3 simulation. Recall that COIHiTrust is built on P2P trust assessment results. Specifically, $T_j^{COI-HiTrust}(t)$ is

obtained from Equation 6 after collecting $T_{ij}(t)'$s from nodes in the same subtask groups at time *t*, assuming $w^{social}/w^{QoS} = 0.5/0.5$. Figure 4 plots trust bias (the difference between $T_j^{OBJ}(t)$ and $T_j^{COI-HiTrust}(t)$) over time for a trustee node (node *j*) randomly picked. There are several curves in Figure 4, each corresponding to a different compromise rate λ_{com}. Again the solid lines are analytical results and the dashed lines are ns-3 simulation results. Figure 4 confirms trust accuracy and convergence behavior of COIHiTrust. We observe that the trust bias is well under control, i.e., it is less than 0.02 for low compromise rates and less than 0.08 for high compromise rates.

5.3 Effect of Uncertainty and Noise

Noise and uncertainty is modeled by $TE_{ij}^{X,direct}(t)$ defined by Equation 3. Figure 5 plots $TB_{ij}^{honesty}(t)$ over time for a node randomly picked. However, instead of setting $TE_{ij}^{honesty,direct}(t) = 0$, $TE_{ij}^{honesty,direct}(t)$ is a random variable following uniform distribution U[0, 0.2] so that the trust error of direct honesty trust assessment to go as high as 0.2. We see that the trend exhibited in Figure 5 is remarkably similar to that of Figure 3. $TB_{ij}^{honesty}(t)$ in Figure 5 is about 0.1 higher than its counterpart in Figure 3 because the average value of $TE_{ij}^{honesty,direct}(t)$ is 0.1, given that it follows U[0, 0.2] distribution. This verifies that our trust propagation and aggregation protocol design (Equations 1 and 2) is resilient to $TE_{ij}^{X,direct}(t)$ since there is no extra trust error being introduced during trust propagation and aggregation. Our protocol's resiliency is attributed to its ability to adjust the best set of ($\alpha^{honesty}$, $\beta^{honesty}$) values dynamically in response to noise detected at runtime.

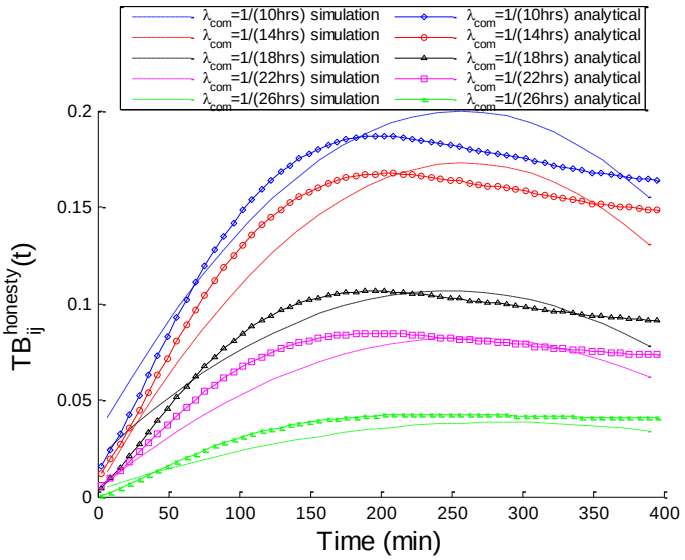

Figure 5: $TB_{ij}^{honesty}(t)$ vs. t with noise in U[0, 0.2].

VI. APPLICATION: MISBEHAVING NODE DETECTION

We propose a novel scheme to utilize COI-HiTrust for IDS functionality for the misbehaving node detection application. The basic idea is to have the SGL (or the commander) make a decision periodically whether a node (or a SGL) is considered untrustworthy or compromised based on the peer-to-peer trust evaluation results sent to it. The IDS strategy we investigate is as follows: when a node's trust level falls below the system minimum trust threshold, say, T^{th}, the node is diagnosed as completely untrustworthy and thus compromised. The IDS formed is characterized by its false positive probability, P_{fp}^{IDS}, i.e., the probability of misdiagnosing a good node as a bad node, and false negative probability, P_{fn}^{IDS}, i.e., the probability of misdiagnosing a bad node as a good node. With the help of the semi-Markov stochastic processes developed, we can fairly accurately predict P_{fp}^{IDS} and P_{fn}^{IDS}. More specifically, we leverage the knowledge of whether a node is compromised or not at time t from the semi-Markov stochastic process model to predict $P_{fp}^{IDS}(t)$ and $P_{fn}^{IDS}(t)$ obtainable. Suppose that in a subtask group with n + 1 nodes, each node, say i ($i \neq j$), reports its peer-to-peer trust evaluation result $T_{i,j}(t)$ to the SGL. Based on our IDS strategy if the expected trust value of node j at time t, $\mu_j(t)$, is below T^{th}, the SGL will consider node j as totally untrustworthy and thus compromised. Suppose that the peer-to-peer trust value toward node j is a random

variable following *t*-distribution and thus the SGL has $nT_{i,j}(t)$ sample values collected from *n* nodes in the same subtask group. Then, we will have a random variable $X_j(t)$ with *n*-1 degree of freedom, i.e.,

$$X_j(t) = \frac{\overline{T_{i,J}(t)} - \mu_j(t)}{S_j(t)/\sqrt{n}} \tag{26}$$

where $\overline{T_{iJ}(t)}$ and $S_j(t)$ are the sample mean and sample standard deviation of node *j*'s trust value, respectively. Thus, the probability that node *j* is judged as a compromised node at time *t* is:

$$\begin{aligned} \Theta_j(t) &= \Pr\left(\mu_j(t) < T^{th}\right) \\ &= \Pr\left(X_j(t) > \frac{\overline{T_{i,J}(t)} - T^{th}}{S_j(t)/\sqrt{n}}\right) \end{aligned} \tag{27}$$

The anticipated false positive probability at time *t* can be obtained by calculating $\Theta_j(t)$ under the condition that node *j* is not compromised. Similarly, the false negative probability at time *t* can be obtained by calculating $1 - \Theta_j(t)$ under the condition that node *j* is compromised. That is,

$$P_{fp,j}^{IDS}(t) = \Pr\left(X_j(t) > \frac{\overline{T_{ij}^{N}(t)} - T^{th}}{S_j^{N}(t)/\sqrt{n}}\right) \tag{28}$$

$$P_{fn,j}^{IDS}(t) = \Pr\left(X_j(t) \leq \frac{\overline{T_{ij}^{C}(t)} - T^{th}}{S_j^{C}(t)/\sqrt{n}}\right) \tag{29}$$

Here $\overline{T_{ij}^{N}(t)}$ and $S_j^{N}(t)$ are the mean value and standard deviation of node *j*'s trust value reported by all nodes in the same subtask group, conditioning on node *j* not having been compromised at time *t*, while $\overline{T_{i,j}^{C}(t)}$ and $S_j^{C}(t)$ are the mean value and standard deviation of node *j*'s trust value, conditioning on node *j* having been compromised at time *t*. Note that only Equation 26 will be used by a SGL (or a commander) based on *n* $T_{i,j}(t)$ values collected at time *t* to judge if node *j* is totally untrustworthy or compromised. Equations 28 and 29 are used to predict the resulting false positive probability $P_{fp,j}^{IDS}(t)$ and false negative probability $P_{fn,j}^{IDS}(t)$, given the knowledge whether node *j* is actually compromised or not at time *t*, which we can easily find out from the mathematical model output.

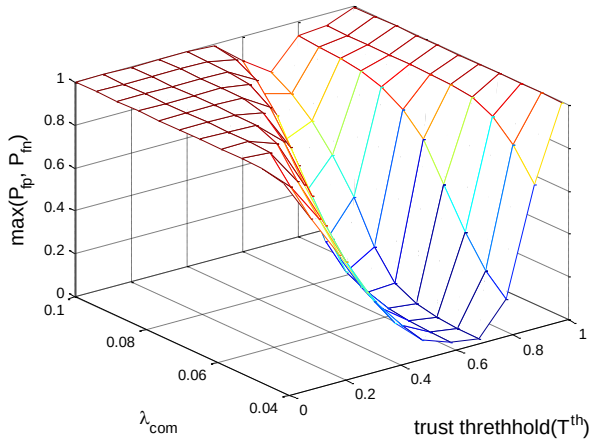

Figure 6: Effect of T^{th} and λ_{com} on $max(P_{fp}, P_{fn})$.

For the intrusion detection application, the application-level parameters for *application performance optimization* include (a) the drop-dead minimum trust threshold T^{th} and (b) the weights w^{social} and w^{QoS} associated with social trust and QoS trust, with $w^{social} + w^{QoS} = 1$.

6.1 Sensitivity Analysis of the Drop-Dead Minimum Trust Threshold T^{th}

Figure 6 shows $max(P_{fp}, P_{fn})$ vs. the compromise rate λ_{com} and the minimum trust threshold T^{th} as a result of executing the trust-based intrusion detection application, where P_{fp} and P_{fn} are the time-averaged false positive and false negative probabilities calculated from Equations 28 and 29, respectively, over all nodes in the system. Here $w^{social} = w^{QoS} = 0.5$ to isolate its effect. $max(P_{fp}, P_{fn})$ is used as the performance metric because there is a tradeoff between P_{fp} and P_{fn}. That is, as the minimum trust threshold T^{th} increases, the false negative probability P_{fn} decreases while the false positive probability P_{fp} increases. We see that given a compromise rate λ_{com} value for trust formation, there exists an optimal trust threshold T^{th} at which $max(P_{fp}, P_{fn})$ is minimized. Further, we can visually observe the effect of our proposed *application performance maximization* design in this intrusion detection application. Specifically, Figure 6 identifies that the optimal T^{th} value is 0.6 when λ_{com}= 0.05 to minimize P_{fp} without penalizing P_{fn} but the optimal T^{th} value increases to 0.7 as λ_{com} increases to 0.1 so as to minimize P_{fn} without compromising P_{fp}, thus minimizing $max(P_{fp}, P_{fn})$ as a result.

6.2 Sensitivity Analysis of Trust Formation (w^{social} and w^{QoS})

Figure 7 shows $max(P_{fp}, P_{fn})$ vs. the minimum trust threshold T^{th} and the weight w^{social} associated with social trust (with $w^{QoS} = 1 - w^{social}$). Here λ_{com}=0.05 to isolate its effect. Maximizing application performance, i.e., minimizing $max(P_{fp}, P_{fn})$, is achieved by adjusting w^{social} to allow trust to be formed out of the best combination of social trust and QoS trust components. We see that the best w^{social} value is 0.5 when T^{th} is set at 0.6 to minimize $max(P_{fp}, P_{fn})$. However, $max(P_{fp}, P_{fn})$ is minimized among all when w^{social} is 0.9 and T^{th} is 0.5. As the environment conditions change dynamically, e.g., as λ_{com} changes from 0.05 to 0.1, there exists the best combination of w^{social} and T^{th} values that will maximize the application performance in terms of minimizing $max(P_{fp}, P_{fn})$.

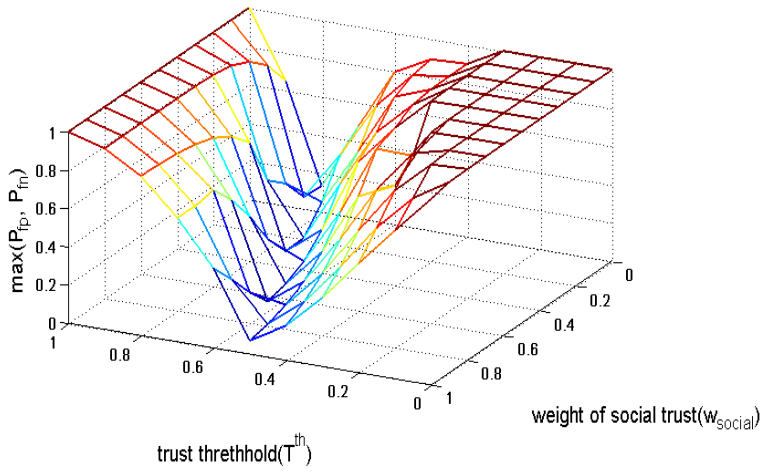

Figure 7: Effect of T^{th} and w_{social} on $max(P_{fp}, P_{fn})$.

VII. APPLICABILITY

The identification of optimal protocol settings in terms of (α^X, β^X) to minimize trust bias, and the best application-level trust optimization settings in terms of w^{social} and T^{th} to maximize application performance, i.e., minimizing $max(P_{fp}, P_{fn})$ for the intrusion detection application is performed at static time. One way to apply the results for dynamic hierarchical trust management is to build a lookup table at static time listing the optimal protocol settings discovered over a perceivable range of parameter values. Then, at runtime, upon sensing the environment conditions matching with a set of parameter values, a node can perform a simple table lookup operation augmented with extrapolation/interpolation techniques to determine and apply the

optimal protocol setting to minimize trust bias and maximize application performance dynamically in response to environment changes. The complexity is O(1) because of the table lookup technique employed.

VIII. CONCLUSION

In this paper, we designed and analyzed a dynamic hierarchical trust management protocol for managing community of interest mobile groups in heterogeneous MANET environments. We demonstrated desirable resiliency and accuracy properties of our protocol design by means of a novel model-based analysis methodology with simulation validation. We also demonstrated its utility with a misbehaving node detection application built on top of our protocol based on a new design concept of mission-dependent trust formation for achieving application performance maximization. We proposed an efficient table-lookup method for applying the analysis results at runtime dynamically in response to changing environment conditions to maximize application performance in terms of minimizing the false alarm rate.

In the future, we plan to consider more sophisticated attacker models such as random, opportunistic, and insidious attacks [15, 29, 30] to further test the resiliency of our hierarchical trust management protocol design and extend the analysis to the Internet of things systems and cyber physical systems where hierarchical control is essential for achieving scalability, reconfigurability, survivability and intrusion tolerance.

ACKNOWLEDGMENT

This material is based upon work supported in part by the U. S. military Research Laboratory and the U. S. military Research Office under contract number W911NF-12-1-0445.

REFERENCES

[1] P.B. Velloso, R.P. Laufer, D. de Cunha, O.C. Duarte, and G. Pujolle, "Trust management in mobile ad hoc networks using a maturity-based model," *IEEE Trans. on Network and Service Management*, vol. 7, no. 3, Sep. 2010, pp. 172-185.

[2] H. Yu, M. Kaminsky, P.B. Gibbons, and A.D. Flaxman, "SybilGuard: Defending Against Sybil Attacks via Social Networks," *IEEE/ACM Transactions on Networking,* vol. 16, no. 3, June 2008, pp. 576-589.

[3] M. Maheswaran, H. C. Tang, and A. Ghunaim, "Toward a Gravity-based Trust Model for Social Networking Systems," *27th Int'l Conf. on Distributed Computing Systems Workshops*, June 2007, pp. 24-31.

[4] E.M. Daly and M. Haahr, "Social Network Analysis for Information Flow in Disconnected Delay-Tolerant MANETs," *IEEE Transactions on Mobile Computing*, vol. 8, no. 5, May 2009, pp. 606-621.

[5] S. Kosta, A. Mei, and J. Stefa, "Small World in Motion (SWIM): Modeling Communities in Ad-Hoc Mobile Networking," *7th IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Boston, MA, USA, 2010.

[6] Q. Li, S. Zhu, and G. Cao, "Routing in Socially Selfish Delay Tolerant Networks," *IEEE Conference on Computer Communications*, San Diego, CA, 2010, pp. 1-9.

[7] G. Ciardo, R.M. Fricks, J.K. Muppala and K.S. Trivedi, *Stochastic Petri Net Package (SPNP) Users Manual*, Department Electrical Engineering, Duke University, 1999.

[8] J.H. Cho, I.R. Chen and D.C. Wang, "Performance Optimization of Region-based Group Key Management in Mobile Ad Hoc Networks," *Performance Evaluation*, vol. 65, no. 5, 2008, pp. 319-344.

[9] J. H. Cho, A. Swami and I. R. Chen, "Modeling and Analysis of Trust Management for Cognitive Mission-driven Group Communication Systems in Mobile Ad Hoc Networks," *International Conference on Computational Science and Engineering*, Vancouver, Canada, Aug. 2009.

[10] J.H. Cho, A. Swami and I.R. Chen, "A Survey on Trust Management for Mobile Ad Hoc Networks," *IEEE Communications Surveys & Tutorials*, Vol. 13, No. 4, Nov. 2011, pp. 562-583.

[11] A. Josang and S. Pope, "Semantic Constraints for Trust Transitivity," *Proc. 2nd Asia-Pacific Conf. on Conceptual Modeling*, Newcastle, Australia, 2005.

[12] J.H. Cho, and I.R. Chen, "Performance Analysis of Hierarchical Group Key Management integrated with Adaptive Intrusion Detection in Mobile Ad Hoc Networks," *Performance Evaluation*, vol. 68, no. 1, 2011, pp. 58-75.

[13] A. daSilva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. Wong, "Decentralized Intrusion Detection in Wireless Sensor Networks," *ACM 1st International Workshop on Quality of Service and Security in Wireless and Mobile Networks*, Montreal, Quebec, Canada, 2005.

[14] F. Bao, I.R. Chen, M. Chang, and J.H. Cho, "Hierarchical Trust Management for Wireless Sensor Networks and Its Applications to Trust-Based Routing and Intrusion Detection," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, 2012, pp. 161-183.

[15] R. Mitchell and I. R. Chen, "Effect of Intrusion Detection and Response on Reliability of Cyber Physical Systems," IEEE Transactions on Reliability, vol. 62, no. 1, pp. 199-210, 2013.

[16] R.R.S. Verma, D. O'Mahony, and H. Tewari, "NTM - Progressive Trust Negotiation in Ad Hoc Networks," *IEI/IEE Symposium on Telecommunications Systems Research*, Dublin, Ireland, Nov. 2001, pp. 1-8.

[17] C.R. Davis, "A Localized Trust Management Scheme for Ad Hoc Networks," *International Conference on Networking*, 2004, pp. 671-675.

[18] The ns-3 Network Simulator, Nov. 2011, http://www.nsnam.org/.

[19] X. Li, F. Zhou, and J. Du, "LDTS: A Lightweight and Dependable Trust System for Clustered Wireless Sensor Networks," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, 2013, pp. 924-935.

[20] J.H. Cho, A. Swami and I.R. Chen, "Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks," *Network and Computer Applications*, vol. 35, 2012, pp. 1001-1012.

[21] R. Mitchell, and I.R. Chen, "A Survey of Intrusion Detection in Wireless Networks," *Computer Communications*, vol. 42, April 2014, pp. 1-23.

[22] J. Huang and D. Nicol, "A calculus of trust and its application to PKI and identity management," in *ACM 8th Symposium on Identity and Trust on the Internet*, Gaithersburg, MD, April 2009.

[23] I.R. Chen, F. Bao, M. Chang, and J. H. Cho, "Trust management for encounter-based routing in delay tolerant networks," *IEEE Global Telecommunications Conference*, Miami, FL, Dec. 2010.

[24] I. R. Chen, F. Bao, M. Chang, and J.H. Cho, "Trust-based intrusion detection in wireless sensor networks," *IEEE International Conference on Communications*, Kyoto, Japan, June 2011.

[25] J. Zhang, et al., "A trust management architecture for hierarchical wireless sensor networks," *35th IEEE International Conference on Local Computer Networks*, Denver, Colorado, Oct. 2010, pp. 264–267.

[26] A. Jøsang, and R. Ismail, "The Beta Reputation System," Bled Electronic Commerce Conference, Bled, Slovenia, 2002, pp. 1-14.

[27] E. Ayday, H. Lee, and F. Fekri, "An Iterative Algorithm for Trust Management and Adversary Detection for Delay Tolerant Networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 9, Sept. 2012, pp. 1514-1531.

[28] I.R. Chen, and N. Verma, "Simulation study of a class of autonomous host-centric mobility prediction algorithms for wireless cellular and ad hoc networks," *36th Annual Symposium on Simulation*, pp. 65-72, 2003.

[29] I.R. Chen, A.P. Speer, and M. Eltoweissy, "Adaptive Fault-Tolerant QoS Control Algorithms for Maximizing System Lifetime of Query-Based Wireless Sensor Networks," *IEEE Trans. on Dependable and Secure Computing*, vol. 8, no. 2, 2011, pp. 161-176.

[30] H. Al-Hamadi and I.R. Chen, "Redundancy Management of Multipath Routing for Intrusion Tolerance in Heterogeneous Wireless Sensor Networks," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, 2013, pp. 189-203.

[31] I.R. Chen, and D.C. Wang, "Analyzing Dynamic Voting using Petri Nets," *15th IEEE Symposium on Reliable Distributed Systems*, Niagara Falls, Canada, 1996, pp. 44-53.

[32] I.R. Chen, O. Yilmaz, and I.L. Yen, "Admission control algorithms for revenue optimization with QoS guarantees in mobile wireless networks," *Wireless Personal Communications*, vol. 38, no. 3, pp. 357-376, 2006.

[33] S.T. Cheng, C.M. Chen, and I.R. Chen, "Dynamic quota-based admission control with sub-rating in multimedia servers," *Multimedia Systems*, vol. 8, no. 2, pp. 83-91, 2000.

[34] I. R. Chen, T.M. Chen, and C. Lee, "Performance evaluation of forwarding strategies for location management in mobile networks," *The Computer Journal*, vol. 41, no. 4, 1998, pp. 243–253.

[35] B. Gu, and I. R. Chen, "Performance Analysis of Location-Aware Mobile Service Proxies for Reducing Network Cost in Personal Communication Systems," *Mobile Networks and Application*s, vol. 10, no. 4, 2005, pp. 453–463.

[36] A. A. Selçuk, E. Uzun, and M. R. Pariente, "A Reputation-based Trust Management System for P2P Networks," *Network Security*, vol.6, no.3, May 2008, pp. 235-245.

[37] L. Xiong, and L. Liu, "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities," *IEEE Trans. on Knowledge and Data Engineering*, v.16, pp. 843-857, July 2004.

[38] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in P2P networks," *12th International Conference on World Wide Web*, Budapest, Hungary, May 2003.

[39] Z. Su et al., "ServiceTrust: Trust Management in Service Provision Networks," *IEEE International Conference on Services Computing*, Santa Clara, CA, 2013.

[40] I. R. Chen, F. Bao, M. Chang, and J.H. Cho, "Dynamic Trust Management for Delay Tolerant Networks and Its Application to Secure Routing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, 2014, pp. 1200-1210.

[41] I. R. Chen and J. Guo, "Dynamic Hierarchical Trust Management of Mobile Groups and Its Application to Misbehaving Node Detection," *28th IEEE International Conference on Advanced Information Networking and Applications*. Victoria, Canada, May 2014.