Dynamic Hierarchical Trust Management of Mobile Groups and Its Application to Misbehaving Node Detection

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

Abstract— **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. We combine the designs of *COI hierarchical management* for scalability and reconfigurability with *COI dynamic trust management* for survivability and intrusion tolerance to compose a scalable, reconfigurable, and survivable COI management protocol for managing COI mission-oriented mobile groups in heterogeneous mobile environments. A COI mobile group in this environment would consist of heterogeneous mobile entities such as communication-device-carried personnel/robots and aerial or ground vehicles operated by humans exhibiting not only quality of service (QoS) characters, e.g., competence and cooperativeness, but also social behaviors, e.g., connectivity, intimacy and honesty. A COI commander or a subtask leader must measure trust with both social and QoS cognition depending on mission task characteristics and/or trustee properties to ensure successful mission execution. 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*— *Trust management; community of interest; scalability; adaptability; intrusion detection; performance analysis.

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, as well as secret key rekeying operations to maintain the hierarchical structure and to ensure secure group communication functionality.

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 such as manned or unmanned, and subtask group characteristics such as 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 movement 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 extending from our preliminary work on trust management for mobile ad hoc and sensor networks (MANETs) [4, 9, 14]. 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 [10, 11]. However there is very little research on hierarchically structured trust management protocol design for MANETs. Verma et al. [15] and Davis [16] considered hierarchical trust management for MANETs. In their hierarchical trust management schemes, each node performs trust evaluation locally. However, their schemes heavily rely on the certificates issued off-line or by trusted third parties which typically are not available in MANET environments.

We envision the following original contributions from this work:

1. Unlike most existing reputation and trust management schemes in the literature [10, 11], 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 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. Untreated in the literature, 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. For trust formation, we explore a new design concept of *mission-dependent trust formation* with the goal of *application performance optimization*, allowing trust being formed out of social and QoS trust properties. Dynamic trust management 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. We use a *misbehaving node detection* application as an example for which we identify the best application-level drop-dead trust threshold below which a node is considered misbehaving, and that the minimum trust threshold can be adjusted in response to changing conditions to minimize the false alarm probability.
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.

II. System Model and 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. 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/deregistration actions to the SGLs.

We assume that for security reasons, the COI group uses a group key for communication. When a node joins or leaves the COI group, the group key is rekeyed immediately to satisfy the secrecy requirement. Various key generation schemes may be used for this purpose. We assume that a key generation protocol such as the Group Diffie-Hellman (GDH) protocol [8] can be used for group key generation to ensure secure group communication against outside attackers. However, it does not provide tolerance against compromised nodes (or inside attackers) who know the group key.

Our solution is COI-HiTrust 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* [2-3], 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 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.

We assume the presence of malicious and selfish nodes. A selfish node may act uncooperatively, the degree of which depends on whom it works with 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. In particular, it can perform the following trust-related attacks to disrupt the trust system:

Self-promoting attacks: it can promote its importance by providing good recommendations for itself to improve its trust status.

Bad-mouthing attacks: it can ruin the reputation of well-behaved nodes by providing bad recommendations against good nodes.

Ballot stuffing attacks: it can boost the reputation of bad nodes by providing good recommendations for them for collusion.

COI-HiTrust is said to be resilient to the above *trust-related attacks* when the "*subject 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.

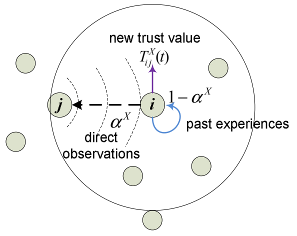

Figure 1(a): Node *i* evaluates node *j* with direct observations and past experiences in trust property *X*.

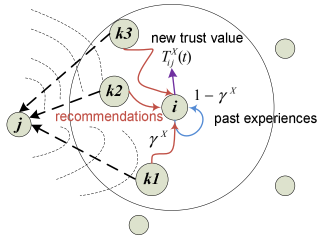

Figure 1(b): Node *i* evaluates node *j* with recommendations and past experiences in trust property *X*.

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 or two SGLs in a COI.

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:

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

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:

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

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 Evidence-Based Trust Aggregation

In Equation 1 there is a direct-observation 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*. If there is a priori knowledge that node *i* is close to node *j,* e.g., deriving from a "friendship" matrix as input, then $T_{i,j}^{intimacy,direct}(t) = 1$. Otherwise node *i* can compute $T_{i,j}^{intimacy,direct}(t)$ 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, $T_{i,j}^{intimacy,direct}(t)$ can be 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 $[t - d\Delta t, t]$. Node *i* estimates $T_{i,j}^{honesty,direct}(t)$ by the ratio of the number of suspicious interaction experiences observed during $[t - d\Delta t, t]$ to a system honesty threshold to reduce false positives. Node *i* can use a set of anomaly detection rules such as interval, retransmission, jamming and delay rules [13] to keep a count of suspicious experiences of node *j* during $[\, t - d\Delta t, t\,]$. If the count exceeds the honesty threshold, node *i* considers node *j* as totally dishonest, i.e., $T_{i,j}^{honesty,direct}(t)$=0. Otherwise it is equal to 1 less the ratio of the count to the threshold. Alternatively keeping the count, we can also use the beta distribution as commonly used in Bayesian Inference to derive $T_{i,j}^{honesty,direct}$.
- $T_{i,j}^{unselfishness,direct}(t)$: This provides the belief of node *i* that node *j* is unselfishness based on direct observations during $[t - d\Delta t, t]$. Node *i* can estimate $T_{ij}^{unselfishness,direct}(t)$ 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*'s is competent at time *t*. Node *i* can overhear 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. Node *i* 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:

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

$TE_{ij}^{X,direct}(t)$ above is one source of trust inaccuracy. Another source of trust inaccuracy is due to compromised nodes

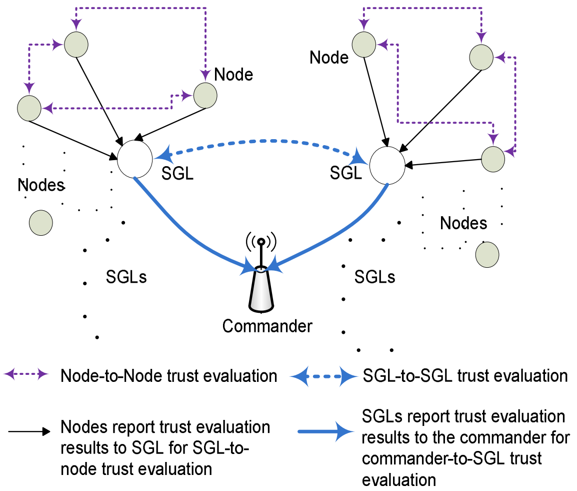
Node
Node
Nodes
SGL
SGL
Nodes
SGLs
SGLs
Commander
Node-to-Node trust evaluation
SGL-to-SGL trust evaluation
Nodes report trust evaluation results to SGL for SGL-to-node trust evaluation
SGLs report trust evaluation results to the commander for commander-to-SGL trust evaluation

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

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:

$$TB_{ij}^{X}(t) = T_{ij}^{X}(t) - T_{j}^{X}(t) \qquad (4)$$

$TB_{ij}^{X}(t)$ 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, 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,

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

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 $T_{ij}^{social}(t) = 0.5\, T_{ij}^{intimacy}(t) + 0.5\, T_{i,j}^{honesty}(t)$.

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,

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

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 the *Misbehaving Node Detection* section 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. 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 [6, 7, 20-24] 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 for movements between subtask groups, compromise rate, selfish rate, hardware/software failure rate, initial energy, 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 6) to be compared with subjective trust $T_j^{COI-HiTrust}(t)$ obtained in Equation 7 for accuracy assessment.

Table 1 lists the default parameter values. We consider an environment with *N*=400 heterogeneous mobile 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. Physically, nodes move according to the SWIM(speed, pause) 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 (except for the commander) 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-promotion, bad-mouthing and ballot-stuffing attacks as described in Section 2. 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.

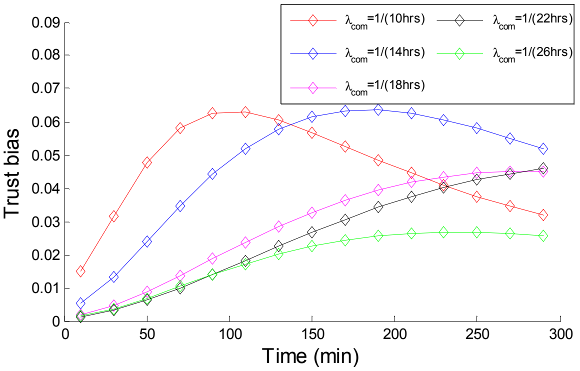

Figure 3: $TB_{ij}^{honesty}(t)$ over time.

Table 1: Parameters and default value/range used.

Parameter	Value	Parameter	Value
E	U[12,24]hrs	*R*	250m
speed	1.45	*N*	400
pause	2 hrs	*d*	2
λ_{com}	1/18hrs	Δt	0.2hr
$\lambda_{selfish}$	1/36hrs	$TE_{ij}^{X,d}$	U[0.0,0.2]

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

4.1 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 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 0.06 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.

4.2 COI-HiTrust Accuracy and Convergence Behavior

In this section, we examine the trust convergence and accuracy behavior of COIHiTrust. 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}. Figure 4 confirms COIHiTrust trust accuracy and convergence behavior. We observe that the trust bias is well under control, i.e., it is less than 0.01 for low compromise rates and less than 0.05 for high compromise rates.

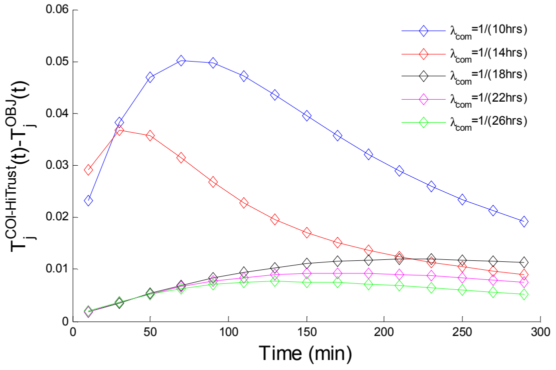

Figure 4: Trust bias of COIHiTrust over time.

4.3 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, 0.2]. We see that the trend exhibited in Figure 5 is remarkably similar to that of Figure 3 despite the presence of noise and uncertainty which can cause the trust error of direct honesty trust assessment to go as high as 0.2. 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.

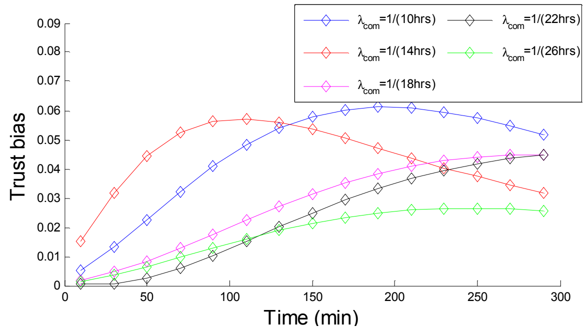

Figure 5: $TB_{ij}^{honesty}(t)$ vs. *t* with noise in U[0, 0.2].

V. 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* ≠ *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 *n* $T_{i,j}(t)$ 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.,

$$X_j(t) = \frac{\overline{T_{\imath,\jmath}(t)} - \mu_j(t)}{S_j(t)/\sqrt{n}} \qquad (7)$$

where $\overline{T_{\imath\jmath}(t)}$ 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:

$$\Theta_j(t) = \Pr\left(\mu_j(t) < T^{th}\right)$$
$$= \Pr\left(X_j(t) > \frac{\overline{T_{\imath,\jmath}(t)} - T^{th}}{S_j(t)/\sqrt{n}}\right) \qquad (8)$$

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,

$$P_j^{fp}(t) = \Pr\left(X_j(t) > \frac{\overline{T_{\imath\jmath}^N(t)} - T^{th}}{S_j^N(t)/\sqrt{n}}\right) \qquad (9)$$

$$P_j^{fn}(t) = \Pr\left(X_j(t) \le \frac{\overline{T_{\imath\jmath}^C(t)} - T^{th}}{S_j^C(t)/\sqrt{n}}\right) \qquad (10)$$

Here $\overline{T_{\imath\jmath}^N(t)}$ 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 $\overline{T_{\imath,\jmath}^C(t)}$ 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 6 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 9 and 10 are used to predict the resulting $P_{fp}^{IDS}(t)$ and $P_{fn}^{IDS}(t)$, 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.

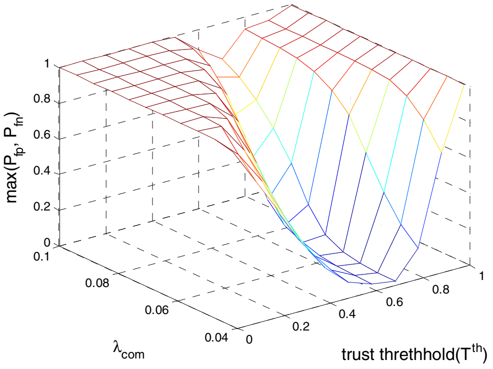

Figure 6: Effect of T^{th} and λ_{com} on $max(P_{fp}, P_{fn})$.

Figure 6 shows $max(P_{fp}, P_{fn})$ vs. λ_{com} and 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 9 and 10, respectively, over all nodes in the system. Here $max(P_{fp}, P_{fn})$ 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^{tt} value is 0.6 when λ_{com}= 0.05 to minimize P_{fp} without penalizing P_{fn} but the optimal T^{tt} value increases to 0.7 as λ_{com} increases to 0.1 so as to minimize P_{fn} without compromising P_{fp}, and, as a result, minimize $max(P_{fp}, P_{fn})$.

VI. Conclusion

In this paper, we designed and analyzed a dynamic hierarchical trust management protocol for managing community of interest mobile groups in heterogeneous mobile systems. We demonstrated desirable resiliency and accuracy properties of our protocol design by means of a novel model-based analysis methodology. 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. In the future, we plan to consider more sophisticated attacker models such as random, opportunistic, and insidious attacks [17, 18, 19] to further test the resiliency of our trust protocol design.

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] 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.

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

[6] I.R. Chen, and D.C. Wang, "Analysis of Replicated Data with Repair Dependency," *The Computer Journal*, vol. 39, no. 9, 1996, pp. 767-779.

[7] 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.

[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," *Inter. Conf. Computational Science and Engineering,* Vancouver, Canada, 2009, pp. 641-650.

[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] K. Govindan and P. Mohapatra, "Trust Computations and Trust Dynamics in Mobile Ad hoc Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, 2012, pp. 279-298.

[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. 67, 2010.

[13] A. daSilva, et al., "Decentralized Intrusion Detection in Wireless Sensor Networks," *ACM 1st 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.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, 2001, pp. 1-8.

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

[17] 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.

[18] 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.

[19] 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.

[20] Y. Li and I.R. Chen, "Design and performance analysis of mobility management schemes based on pointer forwarding for wireless mesh networks," *IEEE Transactions on Mobile Computing,* vol. 10, no. 3, 2011, pp. 349-361.

[21] I.R. Chen and T.H. Hsi, "Performance analysis of admission control algorithms based on reward optimization for real-time multimedia servers," *Performance Evaluation*, vol. 33, no. 2, pp. 89-112, 1998.

[22] 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.

[23] 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.

[24] 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 Applications*, vol. 10, no. 4, 2005, pp. 453–463.