

Hierarchical Trust Management for Wireless Sensor Networks and Its Applications to Trust-based Routing and Intrusion Detection

Fenye Bao, Ing-Ray Chen, MoonJeong Chang, and Jin-Hee Cho

Abstract— We propose a highly scalable cluster-based hierarchical trust management protocol for wireless sensor networks (WSNs) to effectively deal with selfish or malicious nodes. Unlike prior work, we consider multidimensional trust attributes derived from communication and social networks to evaluate the overall trust of a sensor node. By means of a novel probability model, we describe a heterogeneous WSN comprising a large number of sensor nodes with vastly different social and quality of service (QoS) behaviors with the objective to yield “ground truth” node status. This serves as a basis for validating our protocol design by comparing *subjective trust* generated as a result of protocol execution at runtime against *objective trust* obtained from actual node status. To demonstrate the utility of our hierarchical trust management protocol, we apply it to trust-based geographic routing and trust-based intrusion detection. For each application, we identify the best trust composition and formation to maximize application performance. Our results indicate that trust-based geographic routing approaches the ideal performance level achievable by flooding-based routing in message delivery ratio and message delay without incurring substantial message overhead. For trust-based intrusion detection, we discover that there exists an optimal trust threshold for minimizing false positives and false negatives. Furthermore, trust-based intrusion detection outperforms traditional anomaly-based intrusion detection approaches in both the detection probability and the false positive probability.

Index Terms— Trust management; security; wireless sensor networks; routing; intrusion detection; performance analysis.

I. INTRODUCTION

A wireless sensor network (WSN) is usually composed of a large number of spatially distributed autonomous sensor nodes (SNs) to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants. A SN deployed in the WSN has the capability to read the sensed information and transmit or forward information to base stations or a sink node through multi-hop routing. While SNs have popularly used for various monitoring purposes such as wild animals, weather, or environments for battlefield surveillance, they also have severely restricted resources such as energy, memory, and

computational power. Further, wireless environments give more design challenges due to inherently unreliable communication. A more serious issue is that nodes may be compromised and perform malicious attacks such as packet dropping or packet modifications to disrupt normal operations of a WSN wherein SNs usually perform unattended operations. A large number of SNs deployed in the WSN also require a scalable algorithm for highly reconfigurable communication operations.

In this work, we propose a hierarchical trust management protocol leveraging clustering to cope with a large number of heterogeneous SNs for scalability and reconfigurability, as well as to cope with selfish or malicious SNs for survivability and intrusion tolerance. We address the key design issues of trust management including trust composition (i.e., what trust components are considered), trust aggregation (i.e., how information is aggregated for each trust component), and trust formation (i.e., how trust is formed from individual trust components). The scientific contributions of the paper are as follows:

1. Unlike most existing reputation and trust management schemes in the literature [1], we consider not only quality of service (QoS) trust derived from communication networks, but also social trust derived from social networks [2] to judge if a node is trustworthy to deal with selfish (uncooperative) or malicious nodes.

2. Untreated in the literature, we design and validate a hierarchical trust management protocol that can dynamically learn from past experiences and adapt to changing environment conditions (e.g., increasing hostility or misbehaving node population) to maximize application performance and enhance operation agility. This is achieved by addressing critical issues of hierarchical trust management, namely, trust composition, aggregation, and formation. For trust composition, we explore novel social and QoS trust components. For trust aggregation, we identify the best way to aggregate trust (direct vs. indirect trust evaluation) and propagate trust (trust data collection, dissemination and analysis) for *each* individual trust component, and ascertain protocol accuracy by means of a novel model-based analysis methodology. For trust formation, we identify the best way to form trust out of social and QoS trust properties depending on application requirements to maximize application performance. 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., increasing hostility) and then at runtime by learning and adapting to changing environment

Manuscript received April 8, 2011; revised August 25, 2011 and October 27, 2011. The associate editor coordinating the review of this paper and approving it for publication was E. Bertino.

Fenye Bao, Ing-Ray Chen and MoonJeong Chang are with the Department of Computer Science, Virginia Tech, Falls Church, VA 22043; (e-mail: {baofenye, irchen, mjchang}@vt.edu).

Jin-Hee Cho is with Computational and Information Sciences Directorate, U.S. Army Research Laboratory, Powder Mill Rd. Adelphi, MD 20783; (e-mail: jinhee.cho@us.army.mil).

conditions using the best trust formation model identified from static analysis.

3. To achieve the goals of identifying the best trust composition, trust aggregation and trust formation for WSN applications, we develop a novel model-based analysis methodology for analyzing and validating protocol design. The novelty lies in the new design notion of *objective trust* derived from global knowledge or ground truth derived from the mathematical model against which *subjective trust* obtained as a result of executing the trust management protocol may be compared and validated. This requires a mathematical model based on Stochastic Petri Net (SPN) techniques [3] and an iteration solution technique be developed to faithfully describe a large number of heterogeneous mobile entities with a variety of QoS and social behaviors to yield global knowledge or ground truth of node status, thus providing *objective trust* against which *subjective trust* from protocol execution can be validated. The end product is a model-based analysis tool for evaluation of hierarchical trust management protocol designs applicable to a wide range of WSN applications, allowing trust composition, trust aggregation, and trust formation designs to be incorporated, tested and validated.

4. Untreated in the literature, we explore and validate a new design concept of *application-level trust optimization* in response to changing conditions to maximize application performance or best satisfy application requirements. To demonstrate the utility of the hierarchical trust management protocol, we apply it to *trust-based geographic routing* [4, 5] and *trust-based intrusion detection*. For the trust-based geographical routing application, we identify the best trust formation model to optimize application performance in delivery ratio or message delay in the presence of misbehaving nodes. For the trust-based intrusion detection application, we identify the best trust formation model as well as the best application-level drop-dead trust threshold below which a node is considered misbehaving to optimize application performance in false alarm probability.

The rest of the paper is organized as follows. In Section II we survey existing work in trust management for WSNs, as well as trust-based routing and trust-based intrusion detection in WSNs. In Section III, we describe the system model. In Section IV, we describe our hierarchical trust management protocol addressing the issues of trust formation, trust aggregation, and trust composition in a hierarchically structured WSN. In Section V, we develop a probability model to describe a clustered WSN consisting of a large number of nodes with vastly different social and QoS behaviors to yield ground truth node status for validation purposes. In Section VI, we apply the hierarchical trust management protocol to the clustered WSN described in Section V and identify the best *trust aggregation* model for each individual trust component, such that subjective trust obtained as a result of executing the protocol is close to objective trust obtained from ground truth node status. In Section VII, we apply the hierarchical trust management protocol to *trust-based geographic routing* as an application and identify the best trust formation model for optimizing application performance in delivery ratio and delay with

dynamic trust management control. In Section VIII, we apply the hierarchical trust management protocol to *trust-based intrusion detection* with application-level trust optimization as another application, with results and physical interpretations given. Finally in Section IX, we conclude the paper and outline some future research areas.

II. RELATED WORK

In the literature, trust has been used in WSNs for assessing the availability, reliability, or security property of a node (e.g., whether a node is malicious or not) based on past interaction experiences [6-11]. Ganerwal et al. [7] proposed a reputation-based framework for data integrity in WSNs. The proposed reputation system takes information collected by each node using a *Watchdog* mechanism (for direct monitoring and observations) to detect invalid data and uncooperative nodes. Yao et al. [11] proposed a parameterized and localized trust management scheme for WSN security, particularly for secure routing, where each node only maintains highly abstracted parameters to evaluate its neighbors. Aivaloglou and Gritzalis [6] proposed a hybrid trust and reputation management protocol for WSNs by combining *certificate-based* and *behavior-based* trust evaluations. However, [6, 7, 11] cited above only considered a node's QoS property in trust evaluation. Also the analysis was conducted based on a flat WSN architecture which is not scalable. Liu et al. [8] and Moraru et al. [9] proposed trust management protocols and applied them to geographic routing in WSNs. However, no hierarchical trust management was considered for managing clustered WSNs. Their work again evaluated trust based on QoS aspects only such as packet dropping and the degree of cooperativeness, while our work considers both QoS and social trust for trust evaluation of a SN.

Capra et al. [12, 13] discussed the notion of human trust which could be formed from three sources: *direct experiences*, *credentials* and *recommendations*. In particular, recommendations are trust information coming from other nodes in the social context. We consider only two sources in our notion of trust, namely, direct experiences and recommendations, since it is hard for SNs with limited resources to carry credentials. A significant difference of Capra's work from our work is that we specifically consider individual QoS and social trust property, say, X , and devise specific trust aggregation protocols using direct experiences and recommendations to form trust property X , while Capra used the three sources of information to form human trust. Moreover, because different trust properties have their own intrinsic trust nature and react differently to trust decay over time, we identify the best way for each trust property X to take in direct experiences and recommendations information so that the assessment of trust property X would be the most accurate against actual status in trust property X . Another significant difference is that we consider trust formation as the issue of forming the overall "trust" out of individual social and QoS trust properties, while Capra considered it as the issue of forming human trust out of the three sources of trust information. Lastly, we introduce new design concepts of

dynamic trust management and *application-level trust optimization* in response to changing conditions to maximize application performance, and demonstrate the feasibility with trust-based applications, by identifying the best way to form trust as well as use trust out of individual social and QoS trust properties at runtime to optimize application performance.

Shaikh et al. [10] proposed a group-based trust management scheme for clustered WSNs in which each SN performs peer evaluation based on direct observations or recommendations, and each cluster head (CH) evaluates other CHs as well as SNs under its own cluster. This work is similar to ours in that a hierarchical structure is employed for scalability. However, trust in their case is assessed only based on past interaction experiences in message delivery, which in our case is just one possible trust component along with other social and QoS trust components comprising the overall trust metric. Furthermore, we address the trust formation issue (i.e., how a peer-to-peer trust value is formed) to maximize application performance. Zhang et al. [14] followed the same hierarchical trust architecture and considered multi-attribute trust values instead of just one as in [10]. They also considered a decay function that captures the changing nature of trust in trust calculations. 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 address protocol validation issues by devising a mathematical model yielding *objective trust* against which *subjective trust* from protocol execution may be compared for assessing its accuracy.

Intrusion detection is the last defense to cope with malicious nodes for WSNs in which SNs can be compromised due to capture or virus infection. Existing work was mostly based on anomaly detection [15] techniques to discover deviations from expected behaviors, including rule-based [16, 17], weighted summation [18], data clustering [19], and Support Vector Machine (SVM) [20]. In rule-based anomaly detection [16, 17], typically rules based on QoS metrics are being setup to detect suspected attack behaviors, e.g., if a SN does not forward a packet within a time limit, if a SN forwards the same packet multiple times without suppression, or if a packet is received directly from a non-neighbor SN or from a neighbor SN who is not supposed to send a packet during a particular time interval, then the SN in question is suspected of maliciousness. When a SN's "maliciousness count" exceeds a tolerance limit, the SN is diagnosed as compromised. The main drawback of rule-based anomaly detection is that it cannot cope with anomalies not covered by rules, thus leading to high false negatives when unknown anomalies appear. In the weighted summation approach [18], each SN has a weight associated with it representing the trustworthiness of its sensor reading output. The system periodically calculates the average sensor reading output by taking a weighted summation out of all sensor reading outputs. The weight associated with a SN is dynamically updated according to the deviation of the SN's output from the average output. A larger deviation results in a

lower weight. Once the weight of a SN falls below a threshold, the SN is considered a malicious node. The main drawback of this approach is a high false positive probability may result. In the clustering based approach [19], SNs reporting similar sensor reading data out of selected data features are clustered together. Consequently, a SN that does not belong to any cluster or belong to a small cluster is considered an outlier or a compromised SN. The effectiveness of this approach hinges on the accuracy of the underlying clustering algorithm achievable only through heavy learning and computation which may impede its use for real time operation. In SVM-based anomaly detection [20], a kernel function is chosen to map the input data space into a higher-dimensional space. The anomaly detection is formulated as a quadratic optimization problem to find a minimum hyper-sphere that includes the majority of the data points with a certain degree of similarity. The data points that are outside of this hyper-sphere are considered anomalies. However, the challenge of using SVM-based intrusion detection in WSNs is the computational complexity of solving the optimization problem, thus preventing its use for real time operation. A general problem with anomaly detection is high false alarms because noises in wireless transmission may cause uncertainty of information, and limited resources may cause inability to collect accurate and needed information. In this paper, we develop and analyze trust-based intrusion detection and compare its performance with weighted summation [18] and data clustering [19] anomaly detection techniques.

Trust-based intrusion detection has received much attention in the literature because of its elasticity against uncertainty and resiliency against attacks. Wang et al. [21] proposed an intrusion detection mechanism based on trust for mobile ad hoc networks (MANETs). They employed the concepts of evidence chain and trust fluctuation to evaluate a node in the network, with the evidence chain detecting misbehaviors of a node, and the trust fluctuation reflecting the high variability of a node's trust value over a time window. Ebinger et al. [22] introduced a cooperative intrusion detection method also for MANETs based on trust evaluation and reputation exchange. They split the reputation information into trust and confidence for reputation exchanges and then combine them into trustworthiness for intrusion detection. Theodorakopoulos et al. [23] modeled trust evaluation as a path problem and used path semiring and distance semiring operators to combine opinions such that two nodes can establish an indirect trust relation without previous direct interactions. Here we note that most trust-based intrusion detection mechanisms employed for MANETs cannot be directly implemented in WSNs due to limited battery power and resources in SNs. In this paper, we propose hierarchical trust management leveraging clustering to implement light-weight trust-based intrusion detection for WSNs. To the best of our knowledge [1, 24], our work is the first to use trust to implement intrusion detection functionality and evaluate its effectiveness for clustered WSNs.

This work extends from our preliminary work [25] which considered hierarchical trust management for WSNs and its application to trust-based routing, and [26] which considered its application to trust-based intrusion detection. The protocol

design is extended with new design concepts of (a) trust aggregation protocol accuracy, i.e., identifying and validating the best trust aggregation and propagation protocol setting for each individual trust property X such that subjective trust obtained as a result of protocol execution is close to objective trust or ground truth (b) dynamic trust management, i.e., identifying and validating the best way to form trust out of QoS and social trust properties dynamically (in terms of the best weights used for trust properties) in response to changing conditions such as increasing hostility to maximize application performance, and (c) application-level trust optimization, i.e., identifying the best way to use trust for application performance optimization. Both applications (described in Sections VII and VIII) have been substantially extended to demonstrate the feasibility of these new design concepts.

III. SYSTEM MODEL

We consider a cluster-based WSN consisting of multiple clusters, each with a cluster head (CH) and a number of SNs in the corresponding geographical area. CH nodes have more power and resources than SN nodes. The CH in each cluster may be selected based on an election protocol such as HEED [27] at runtime to balance energy consumption vs. CH functionality. A SN forwards its sensor reading to its CH through SNs in the same cluster and the CH then forwards the data to the base station or the destination node (or sink node) through other CHs.

Leveraging this two-level of hierarchy in the WSN, our trust management protocol is conducted using *periodic* peer-to-peer trust evaluation between two SNs and between two CHs. The trust update interval is Δt which is a system design parameter. At the SN level, each SN is responsible to report its peer-to-peer trust evaluation results towards other SNs in the same cluster to its CH which performs CH-to-SN trust evaluation towards all SNs in its cluster. Similarly a CH is responsible to report its peer-to-peer trust evaluation results towards other CHs in the system to the base station which performs station-to-CH trust evaluation towards all CHs in the system. In Section IV, we will describe the protocols for performing peer-to-peer, CH-to-SN and station-to-CH trust evaluations.

Unlike prior work, we compose our trust metric by considering both *social trust* and *QoS trust* to take into account the effect of both aspects of trust on trustworthiness. Social trust in the context of wireless sensors may include intimacy, honesty, privacy, centrality, and connectivity. QoS trust may include competence, cooperativeness, reliability, task completion capability, etc. We formulate our trust protocol such that it is generic and can take a combination of social trust and QoS trust metrics to form the overall trust metric. Without loss of generality, in this work we consider *intimacy* (for measuring closeness based on interaction experiences) and *honesty* (for measuring regularity/anomaly) to measure social trust derived from social networks. We choose *energy* (for measuring competence) and *unselfishness* (for measuring cooperativeness) to measure QoS trust derived from

communication networks. The *intimacy* trust component reflects the relative degree of interaction experiences between two nodes. It follows the maturity model proposed in [28] in that the more positive experiences SN A had with SN B , the more trust and confidence SN A will have toward SN B . The *honesty* trust component strongly implies whether a node is malicious or not. The assumption is that a compromised node is malicious in nature and thus dishonest. Energy is an important metric in WSNs since SNs are extremely constrained in energy. We use energy as a QoS trust metric to measure if a SN is competent in performing its intended function. The unselfishness trust component reflects if a SN can cooperatively execute the intended protocol.

Our trust management protocol can apply to any WSN consisting of heterogeneous SNs with vastly different initial energy levels and different degrees of malicious or selfish behaviors. We apply the trust management protocol to a clustered WSN in which a SN may adjust its behavior dynamically according to its own operational state and environmental conditions. A SN is more likely to become selfish when it has low energy or it has many unselfish neighbor nodes around. Further, a SN is more likely to become compromised when it has more compromised neighbors around. A CH consumes more energy than SNs. After a SN or CH is compromised, it may consume even more energy to perform attacks. On the other hand, a selfish node consumes less energy than an unselfish node as its selfish behavior is reflected by stopping sensing functions and arbitrarily dropping messages.

A compromised SN can perform various attacks including forgery attacks, jamming attacks, Sybil attacks, denial of service attacks, black/sink hole attacks (absorbing and dropping packets), and slandering attacks. Depending on the system failure definition, some of these attacks if successfully performed are fatal. For example if a compromised node uses its shared secret key to perform a forgery attack and the tampered packet reaches the sink node, it can be considered as a system failure as the consequence of the sink node receiving false information may be catastrophic. Thus, the only defense of the system is to quickly detect and evict compromised nodes before a system failure occurs. In this paper, we show that our hierarchical trust management protocol is resilient to black/sink hole attacks and slandering attacks including good-mouthing attacks (recommending a bad node as a good node), and bad-mouthing attacks (recommending a good node as a bad node) in trust-based routing applications (in Section VII). Also our trust management protocol can be effectively applied to implement trust-based intrusion detection (in Section VIII) to deal with other types of attacks.

IV. HIERARCHICAL TRUST MANAGEMENT PROTOCOL

We first describe our hierarchical trust management addressing the problem of trust formation, trust aggregation and trust composition. Later we apply it to the clustered WSN described in the system model to demonstrate its effectiveness.

Our hierarchical trust management protocol maintains two levels of trust: *SN-level* trust and *CH-level* trust. Each SN

evaluates other SNs in the same cluster while each CH evaluates other CHs and SNs in its cluster. 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 SN sends its trust evaluation results toward other SNs in the same cluster to its CH. Each CH performs trust evaluation toward all SNs within its cluster. Similarly, each CH sends its trust evaluation results toward other CHs in the WSN to a ‘‘CH commander’’ which may reside on the base station if one is available, or on a CH elected if a base station is not available. The CH commander performs trust evaluation toward all CHs in the system. The election protocol is outside of the scope of the paper. The reader is referred to HEED [27] for a possible solution.

These two levels of peer-to-peer trust evaluation process consider four different trust components described earlier: intimacy, honesty, energy, and unselfishness. The trust value that node i evaluates towards node j at time t , $T_{ij}(t)$, is represented as a real number in the range of $[0, 1]$ where 1 indicates complete trust, 0.5 ignorance, and 0 distrust. $T_{ij}(t)$ is computed by:

$$T_{ij}(t) = w_1 T_{ij}^{intimacy}(t) + w_2 T_{ij}^{honesty}(t) + w_3 T_{ij}^{energy}(t) + w_4 T_{ij}^{unselfishness}(t) \quad (1)$$

where $w_1, w_2, w_3,$ and w_4 are weights associated with these four trust components with $w_1 + w_2 + w_3 + w_4 = 1$. Deciding the best values of w_1, w_2, w_3 and w_4 to maximize application performance is a trust formation issue which we aim to explore in this paper (see Section VII and Section VIII). Here we note that in the special case in which intimacy and honesty are equally important and energy and unselfishness are also equally important, Equation 1 can be rewritten as $T_{ij}(t) = 0.5 w_{social} [T_{ij}^{intimacy}(t) + T_{ij}^{honesty}(t)] + 0.5 w_{QoS} [T_{ij}^{energy}(t) + T_{ij}^{unselfishness}(t)]$ with $w_{social} + w_{QoS} = 1$.

A. Peer-to-Peer Trust Evaluation

Here we describe how peer-to-peer trust evaluation is conducted, particularly between two peer SNs or two peer CHs. When a trustor (node i) evaluates a trustee (node j) at time t , it updates $T_{ij}^X(t)$ where X indicates a trust component as follows:

$$T_{ij}^X(t) = \begin{cases} (1 - \alpha)T_{ij}^X(t - \Delta t) + \alpha T_{ij}^{X,direct}(t), & \text{if } i \text{ and } j \text{ are 1-hop neighbors;} \\ \text{avg}\{(1 - \gamma)T_{ij}^X(t - \Delta t) + \gamma T_{kj}^{X,recom}(t)\}, & \text{otherwise;} \end{cases} \quad (2)$$

In Equation 2, if node i is a 1-hop neighbor of node j , node i will use its new trust based on direct observations ($T_{ij}^{X,direct}(t)$) and its old trust based on past experiences ($T_{ij}^X(t - \Delta t)$ where Δt is the trust update interval) toward node j to update $T_{ij}^X(t)$. A parameter α ($0 \leq \alpha \leq 1$) is used here to weigh these two trust values and to consider trust decay over time, i.e., the decay of the old trust value and the contribution of the new trust value. A larger α 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 $[0, t]$. Below we describe how each trust component value $T_{ij}^{X,direct}(t)$ can be obtained based on direct observations for the case node i and node j are 1-hop neighbors:

$T_{ij}^{intimacy,direct}(t)$: This measures the level of interaction experiences following the maturity model [28]. It is computed by the number of interactions between nodes i and j over the maximum number of interactions between node i and any neighbor node over the time period $[0, t]$.

$T_{ij}^{honesty,direct}(t)$: This refers to the belief of node i that node j is honest based on node i 's direct observations toward node j . Node i estimates $T_{ij}^{honesty,direct}(t)$ by keeping a count of suspicious dishonest experiences of node j which node i has observed during $[0, t]$ using a set of anomaly detection rules such as a high discrepancy in the sensor reading or recommendation has been experienced, as well as interval, retransmission, repetition, and delay rules as in [17, 29]. If the count exceeds a system-defined threshold, node j is considered totally dishonest at time t , i.e., $T_{ij}^{honesty,direct}(t) = 0$. Otherwise, $T_{ij}^{honesty,direct}(t)$ is computed by 1 minus the ratio of the count to the threshold. An assumption is that a compromised node must be dishonest.

$T_{ij}^{energy,direct}(t)$: This refers to the belief of node i that node j still has adequate energy (representing competence) to perform its intended function. It may be measured by the percentage of node j 's remaining energy. To calculate $T_{ij}^{energy,direct}(t)$, node i estimates node j 's remaining energy by overhearing node j 's packet transmission activities over the time period $[0, t]$, utilizing an energy consumption model as in [30-32].

$T_{ij}^{unselfishness,direct}(t)$: This provides the degree of unselfishness of node j as evaluated by node i based on direct observations over $[0, t]$. Node i can apply overhearing and snooping techniques to detect selfish behaviors of node j such as not faithfully performing sensing and reporting functions, data forwarding functions [10], or the prescribed trust management protocol execution. Node i may give recent interaction experiences a higher priority over old experiences in estimating $T_{ij}^{unselfishness,direct}(t)$. An assumption is that a compromised node must be uncooperative and thus selfish.

On the other hand, if node i is not a 1-hop neighbor of node j , node i will use its past experience $T_{ij}^X(t - \Delta t)$ and recommendations from its 1-hop neighbors ($T_{kj}^{X,recom}(t)$ where k is a recommender) to update $T_{ij}^X(t)$. Node i will only use its 1-hop neighbors (N_i) as recommenders for energy conservation and scalability. If N_i is an empty set, then node i is an orphan in which case $\gamma = 0$ and node i will not be able to contribute to peer-to-peer trust management. The parameter γ is used here to weigh recommendations vs. past experiences and to consider trust decay over time as follows:

$$\gamma = \frac{\beta T_{ik}(t)}{1 + \beta T_{ik}(t)} \quad (3)$$

Here we introduce another parameter $\beta \geq 0$ to specify the impact of “indirect recommendations” on $T_{ij}^X(t)$ such that the weight assigned to indirect recommendations is normalized to $\beta T_{ik}(t)$ relative to 1 assigned to past experiences. Essentially, the contribution of recommended trust increases proportionally as either $T_{ik}(t)$ or β increases. Instead of having a fixed weight ratio $T_{ik}(t)$ to 1 for the special case in which $\beta = 1$, we allow the weight ratio to be adjusted by adjusting the value of β and test its effect on protocol resiliency against slandering attacks such as 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). The recommendation $T_{kj}^{X, recom}(t)$ provided by node k to node i about node j depends on if node k is a good node. If node k is a good node, $T_{kj}^{X, recom}(t)$ is simply equal to $T_{kj}^X(t)$. If node k is a bad node, it can provide $T_{kj}^{X, recom}(t) = 0$ when node j is a good node by means of bad-mouthing attacks, and can provide $T_{kj}^{X, recom}(t) = 1$ when node j is a bad node by means of good-mouthing attacks. In our analysis we assume this worst-case attack behavior to test our protocol resiliency. The new trust value $T_{ij}^X(t)$ obtained from Equation 2 would be the average of the combined trust values of past trust information and recommendations collected at time t .

B. CH-to-SN Trust Evaluation

Each SN reports its trust evaluation toward other SNs in the same cluster to its CH. The CH then applies a generic statistical analysis method (such as Equation 4 below) to $T_{ij}(t)$ values received to perform CH-to-SN trust evaluation towards node j . Further, the CH can also leverage $T_{ij}(t)$ values received to detect if there is any outlier as an evidence of good-mouthing or bad-mouthing attacks. Based on the resulting CH-to-SN trust evaluation result toward node j , the CH determines whether node j is untrustworthy and needs to be excluded from sensor reading and routing duties. Specifically a CH, c , when evaluating a SN, j , will perform intrusion detection by comparing the system minimum trust threshold T^{th} with node j 's trust value, $T_{cj}(t)$, obtained by:

$$T_{cj}(t) = \text{avg}_{i \in M_c \wedge T_{ci}(t) \geq T^{th}} \{T_{ij}(t)\} \quad (4)$$

where M_c is the set of SNs in the cluster. CH c will announce j as compromised if $T_{cj}(t)$ is less than T^{th} ; otherwise, node j is not compromised. Note that we only take into account the trust values received from those SNs which are considered trustworthy by the CH. That is, CH c will take a trust recommendation from node i only if $T_{ci}(t) \geq T^{th}$. Later in Section VIII we will illustrate a statistical analysis methodology to implement trust-based intrusion detection as an application to hierarchical trust evaluation.

C. Station-to-CH Trust Evaluation

Here we first note that the transmission power and capacity of CHs generally are higher than those of SNs. Thus, the

one-hop radio range of CHs is higher than that of SNs. Also a CH after gathering and possibly aggregating sensor readings will forward the information hop-by-hop to the base station through other CHs. Thus, there are a lot of interaction experiences between two neighbor CHs in a WSN, just like two SNs in a cluster. Consequently, CH-to-CH peer evaluation will be conducted in a similar way as SN-to-SN peer evaluation, as discussed in Section IV.A. Each CH reports its trust evaluation toward other CHs in the WSN to the base station which is infallible with physical protection. The CH commander resided on the base station then applies the same statistical analysis method (as in Equation 4) to $T_{ij}(t)$ values received from all CHs in the system to perform station-to-CH trust evaluation towards CH j . The base station determines whether CH j is considered untrustworthy and needs to be excluded from cluster head duties.

V. PERFORMANCE MODEL

We develop a probability model based on stochastic Petri nets (SPN) [3] techniques to describe the behavior of each SN or CH in the WSN described in Section III. It provides a basis for obtaining ground truth status of nodes in the system, thereby allowing us to derive *objective trust* against which subjective trust obtained as a result of executing our hierarchical trust management protocol can be checked and validated. We use SPN as our analytical tool due to its capability to represent a large number of states for complex systems where an underlying model is a semi-Markov or Markov model. Further, we develop a novel iterative hierarchical modeling technique to avoid state explosion problems and to yield efficient solutions.

Figure 1 shows the SPN model that describes the behavior of a SN (or a CH). We consider a heterogeneous WSN consisting of N_{SN} SNs uniformly distributed in an $M \times M$ square-shaped operational area. Each SN is attached to a CH based on its location and so the system will have N_{CH} clusters each with a CH. CHs and SNs have radio range of R and r , respectively. The trust update interval is Δt . Nodes are stationary after the initial deployment.

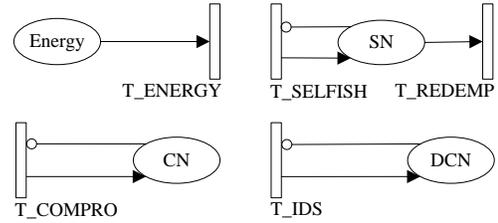


Figure 1: SPN Model for a Sensor Node or a Cluster Head.

Below we explain how we construct the SPN model for describing the behaviors of a single node and how we compose a performance model for the entire WSN using a number of such SPN models (one for each node in the system).

Energy: Place *Energy* indicates the remaining energy level of the node. The initial number of tokens in place *Energy* is set to E_{init} . A token will be released from place *Energy* when transition T_ENERGY is triggered. The rate of transition T_ENERGY indicates the energy consumption rate. A CH consumes more energy than a SN. The energy consumption rate

is also affected by a node's state. It is lower when a node becomes selfish. It is higher when a node is compromised because it takes energy to perform attacks. We denote Δ_{E-SN} , Δ_{E-CH} and $\Delta_{E-compromised}$ as the energy consumption rates per Δt time for a normal SN, a normal CH, and a compromised node, respectively, which can be obtained by analyzing historical data with $\Delta_{E-SN} < \Delta_{E-CH} < \Delta_{E-compromised}$. The energy consumption rates for a selfish SN and a selfish CH are $\rho\Delta_{E-SN}$ and $\rho\Delta_{E-CH}$ per Δt time unit, respectively, with $0 \leq \rho \leq 1$ denoting the energy saving ratio of a selfish node compared with a normal node.

Selfishness: In our WSN system model, a node may become selfish to save energy. A selfish node may stop reading data and drop packets it receives. An unselfish node may turn selfish in every trust evaluation interval Δt according to its remaining energy and the number of unselfish neighbors around. A selfish node may redeem itself as unselfish to achieve a service availability goal when it senses many selfish neighbor SNs around it to balance individual welfare vs. system welfare. We model these behaviors by putting a token into place *SN* when transition $T_{SELFISH}$ is triggered and removing the token from place *SN* when transition T_{REDEMP} is triggered. A token in place *SN* thus indicates that the node is selfish. A node's selfish probability is modeled by:

$$P_{selfish} = \mu \frac{E_{consumed}}{E_{init}} + (1 - \mu) \frac{N_{neighbor}^{unselfish}}{N_{neighbor}} \quad (5)$$

where μ is a weight associated with the *energy* term and $(1 - \mu)$ is the weight associated with the *selfish neighborhood* term. $E_{consumed}$ is energy consumed and E_{init} is the node's initial energy level. Thus, $E_{consumed}/E_{init}$ represents the percentage of energy consumed. $N_{neighbor}^{unselfish}/N_{neighbor}$ is the percentage of unselfish neighbors where $N_{neighbor}^{unselfish}$ is the number of unselfish neighbors and $N_{neighbor}$ is the total number of neighbors. A node's selfish probability tends to be lower when a node has more energy and higher when the node has more unselfish neighbors as there are sufficient unselfish neighbors around to take care of sensor tasks. Thus, the rates of transitions $T_{SELFISH}$ and T_{REDEMP} are given by $P_{selfish}/\Delta t$ and $(1 - P_{selfish})/\Delta t$, respectively. All nodes are unselfish initially with no token in place *SN*. We set μ to 0.5 to give equal weighting to energy and selfish neighborhood terms for the example WSN described in Section III.

Compromise: A node becomes compromised when transition T_{COMPRO} fires and a token is put in place *CN*. The rate to T_{COMPRO} is modeled by:

$$\lambda_c = \lambda_{c-init} \frac{N_{neighbor}^{compromised}}{N_{neighbor}^{uncompromised}} \quad (6)$$

where λ_{c-init} is the initial node compromise rate which can be obtained by first-order approximation based on historical data about the targeted network environment. $N_{neighbor}^{compromised}$ and $N_{neighbor}^{uncompromised}$ are the numbers of compromised and uncompromised nodes in the neighborhood. $N_{neighbor}^{compromised}/N_{neighbor}^{uncompromised}$ refers to the ratio of the

number of compromised 1-hop neighbors to the number of uncompromised 1-hop neighbors. Equation 6 models that a node is more likely to be compromised when there are more 1-hop compromised nodes around it due to collusive attacks. The hierarchically structured WSN has a trust-based intrusion detection system (IDS) in place (see Section VIII). We model the IDS behavior through transition T_{IDS} . A compromised node can be caught by IDS with the rate $(1 - P_{fn})/T_{IDS}$ for transition T_{IDS} where P_{fn} is the IDS false negative probability and T_{IDS} is the IDS detection interval. When a compromised node is detected by the IDS, a token will move to place *DCN*. In addition, we model false positives generated by the IDS (i.e., diagnosing a good node as a bad node) by associating a rate of P_{fp}/T_{IDS} with transition T_{IDS} which is enabled only when the node is not compromised, that is, when there is no token in place *CN*. Note that all nodes are good, i.e., not compromised, initially. Note that trust-based intrusion detection (see Section VIII) will be used for determining IDS P_{fn} and P_{fp} . Also since a compromised node will exhibit uncooperative behaviors (not following the protocol), a compromised node is selfish. This is modeled by moving a token to place *SN* when a token is moved into *CN*. Different from a selfish node, however, a compromised node will not redeem itself to become unselfish again as it is malicious in nature.

The overall performance model for describing the behaviors of a WSN consists of N_{SN} SPN subnet models one for each SN, and N_{CH} SPN subnet models one for each CH, with vastly different energy consumption, selfish/redemption and compromise rates. Below we describe how one could leverage SPN outputs to obtain subjective trust and objective trust values to validate our hierarchical trust management protocol.

A. Subjective Trust Evaluation

Recall that under our proposed trust management protocol, node i will subjectively assess its trust toward node j , $T_{ij}(t)$, based on its direct observations and indirect recommendations obtained toward node j according to Equations 1 and 2. In particular, for the direct trust assessment part when node j is a 1-hop neighbor of node i , node i will apply intimacy, honesty, energy and unselfishness detection mechanisms in the protocol design described in Section IV to assess $T_{ij}^{X,direct}(t)$ based on direct observations over the time period $[0, t]$. Because the assessment is direct, assuming that the detection mechanisms are effective, $T_{ij}^{X,direct}(t)$ computed by node i will be close to actual status of node j at time t , which can be obtained from the SPN model output.

In Table 1, we show how to compute actual status of node j at time t and thus $T_{ij}^{X,direct}(t)$ based on assigning status values to states in the underlying semi-Markov chain of the SPN model, with the state representation of node j being (*Energy*, *CN*, *DCN*, *SN*). Specifically, $T_{ij}^{honesty,direct}(t)$ is approximated by assigning a status value of 0 (representing complete dishonesty) to states in which node j is compromised detected (i.e., *DCN* is 1) and a status value of 1 (representing complete honesty) to all

other states. The reason is that a compromised node must be dishonest. The dishonesty detection mechanisms employed by node i for direct assessment of node j 's dishonesty, however, are at most as good as those employed by the IDS which will announce node j as compromised when it identifies node j as compromised, i.e., when DCN is 1. $T_{ij}^{energy,direct}(t)$ is computed by assigning a status value of $Energy/E_{init}$ to all states. $T_{ij}^{unselfishness,direct}(t)$ is computed by assigning a status value of 1 to states in which node j is not selfish (i.e., SN is 0) and a status value of 0 to states in which node j is selfish (i.e., SN is 1).

Table 1: Status Value Assignments to Compute $T_{ij}^{X,direct}(t)$.

Item	Value	Condition (of node j)
$T_{ij}^{intimacy,direct}(t)$	a/c	If $mark(SN) = 1$ AND $mark(CN) = 0$
	b/c	If $mark(CN) = 1$
	1	Otherwise
$T_{ij}^{honesty,direct}(t)$	1	If $mark(DCN) = 0$
	0	Otherwise
$T_{ij}^{energy,direct}(t)$	$mark(Energy)/E_{init}$	none
$T_{ij}^{unselfishness,direct}(t)$	1	If $mark(SN) = 0$
	0	Otherwise

To compute $T_{ij}^{intimacy,direct}(t)$, we first note that status information in intimacy is not directly available from the state representation. Based on our peer-to-peer trust evaluation protocol (Section IV.A), $T_{ij}^{intimacy,direct}(t)$ is computed by the number of interactions between nodes i and j over the maximum number of interactions between node i and any neighbor node over the time period $[0, t]$. If during the period there is no interaction between nodes i and j , then $T_{ij}^{intimacy,direct}(t) = 0$. Here we predict what $T_{ij}^{intimacy,direct}(t)$ would be when there is a normal level of interactions of data forwarding activities, conditioning on the status of node j , i.e., compromised, selfish or normal. We consider four types of interactions during geographic forwarding, given that node i is the initiating node: (1) *Requesting*: node i broadcasts a packet delivery request to its 1-hop neighbors; (2) *Reply*: nodes that are closer to the destination node than node i will reply to node i ; (3) *Selection*: node i selects up to L nodes with the highest trust values to forward the packet; (4) *Overhearing*: node i overhears if the packet has been forwarded. Node i then keeps track of its interaction experiences with node j to compute $T_{ij}^{intimacy,direct}(t)$. Let the average numbers of interactions of node i with a selfish node, a compromised node and a normal node be a , b and c , respectively. The values of a , b , c are computed dynamically. Below we predict their values from node i 's perspective for the case in which a selfish node drops 50% of packets and a compromised node drops 100% of packets. On the one hand, if node i requests a neighbor to forward a packet then (1) the expected number of interactions between node i and a selfish node j is $25\% \times 50\% \times 3$ because there will be three interactions (reply, selection, and overhearing) only if the selfish node is in the quadrant closest to

the destination node (with 25% probability) and does not drop the packet (with 50% probability); (2) the expected number of interactions between node i and a compromised node j is 0 because a compromised node discards all requests from node i ; and (3) the expected number of interactions between node i and a normal node j is $25\% \times 3$ because there will be three interactions only if that node is in the quadrant closest to the destination node (with 25% probability). On the other hand, if node i receives a request from node j to forward a packet, the expected number of interactions will be $25\% \times 2$ because from node i 's perspective there will be two interactions (reply and selection) only if node i is in the quadrant closest to the target node. Summarizing above, we have:

$$\begin{aligned} a &= 25\% \times 50\% \times 3 + 25\% \times 2; \\ b &= 0 + 25\% \times 2; \\ c &= 25\% \times 3 + 25\% \times 2. \end{aligned} \quad (7)$$

Consequently, we compute $T_{ij}^{intimacy,direct}(t)$ by assigning a status value of a/c to states in which node j is selfish (i.e., SN is 1), b/c to states in which node j is compromised (i.e., CN is 1), and $c/c = 1$ to states in which node j is a normal node ($SN=0$ and $CN=0$).

Here we should emphasize that in practice node i would just follow the protocol execution to assess $T_{ij}^{X,direct}(t)$ using detection mechanisms designed to assess trust property X based on local information. The computational procedure described above is to predict $T_{ij}^{X,direct}(t)$ that would have been obtained by node i based on the argument that a node's direct observation trust assessment would be close to ground truth. Once node i obtains $T_{ij}^{X,direct}(t)$ for $X = \text{honesty, energy, unselfishness and intimacy}$, it will compute $T_{ij}^X(t)$ based on Equation 2 and subsequently $T_{ij}(t)$ based on Equation 1 for subjective trust evaluation.

B. Objective Trust Evaluation

To validate subjective trust evaluation, we compute *objective trust* based on actual status as provided by the SPN model output using exactly the same status value assignment as shown in Table 1 to yield ground truth status of node j at time t . The objective trust value of node j , $T_{j,obj}(t)$, is also a weighted linear combination of four trust component values:

$$\begin{aligned} T_{j,obj}(t) &= w_1 T_{j,obj}^{intimacy}(t) + w_2 T_{j,obj}^{honesty}(t) \\ &+ w_3 T_{j,obj}^{energy}(t) + w_4 T_{j,obj}^{unselfishness}(t) \end{aligned} \quad (8)$$

Note that here $T_{j,obj}^{intimacy}(t)$, $T_{j,obj}^{honesty}(t)$, $T_{j,obj}^{energy}(t)$ and $T_{j,obj}^{unselfishness}(t)$ are objective trust component values, reflecting node j 's ground truth status at time t .

VI. TRUST EVALUATION RESULTS

In this section, we show numerical results obtained through model-based evaluation as described in Section IV. The basis is the example WSN described in Section III characterized by a set of parameter values listed in Table 2. We consider a WSN with 900 SNs (and 81 CHs) evenly spread out in a $900m \times 900m$ operational area based on uniform distribution. The initial

energy lifetime of a SN varies from 360 days to 480 days while the CHs have much higher initial energy lifetime ranging from 720 days to 960 days. The radio ranges of a SN and a CH are $r=50m$ and $R=150m$, respectively. The WSN is assumed to be deployed in a hostile environment with the node's average compromising interval in the range of 80 days to 360 days. We consider the worst case of good-mouthing attacks (providing the highest trust value of 1 for a malicious node) and bad-mouthing attacks (providing the lowest trust value of 0 against a good node). The node is a good node at time $t=0$ and then becomes a bad node based on its compromise rate. The false positive and negative probabilities (P_{fp} and P_{fn}) are in the range of 1% to 5% as a result of trust-based intrusion detection (see Section VIII). Because of the anticipated long system lifetime, to save energy the trust update interval Δt is set to 80 hours. Thus, the amount of energy consumed per Δt time for a normal SN is also set to 80 hours. The amount of energy consumed per Δt time for a normal CH and a compromised node are $\Delta E_{-CH} = 160$ hours and $\Delta E_{-compromised} = 240$ hours, respectively. The energy saving ratio of a selfish node relative to a normal node, ρ , is $1/3$ denoting that a selfish node will only consume energy at $1/3$ of the speed of its unselfish counterpart. After 5-10 runs of trust update from the initial state, the test score quickly stabilizes but does not converge to a single value because we consider a dynamic environment in which trust in energy and honesty is decreasing over time due to energy consumption and node capture.

Table 2: Default Parameter Values Used.

Para	Value	Para	Value	Param	Value
M	900m	R	150m	r	50m
N_{SN}	900	N_{CH}	81	Δt	80hrs
α	[0,1]	β	[0,100]	$1/\lambda_{e-init}$	[80,360]days
ΔE_{-SN}	80hrs	ΔE_{-CH}	160hrs	$\Delta E_{-compromised}$	240hrs
ρ	1/3	T_{IDS}	80hrs	P_{fp}, P_{fn}	[1-5]%
E_{init}	[360,480] days for SNs, [720,960] days for CHs.				

Our trust evaluation consists of two parts. The first part is about trust composition and trust aggregation. The second part is about trust formation. Our assertion is that, because different trust properties have their own intrinsic trust nature and react differently to trust decay over time, each trust property X has its own best α and β values under which subjective assessment of $T_{ij}^X(t)$ from Equation 2 would be the most accurate against actual status of node j in trust property X . Once we are assured of the accuracy of each trust property X , we can then address the trust formation issue for each application in hand, i.e., identifying the best way to form trust out of individual QoS and social trust properties such that the application performance is maximized. We will evaluate trust formation in Section VII and Section VIII when we apply hierarchical trust management to trust-based geographic routing and trust-based intrusion detection.

Recall that a higher α value indicates that subjective trust evaluation relies more on direct observations compared with past experiences while a higher β value indicates that subjective trust evaluation relies more on indirect recommendations provided by the recommenders compared with past

experiences. Below we present CH-to-SN trust evaluation results based on peer-to-peer trust evaluation results reported by SNs in the same cluster, and compare them against objective trust evaluated based on the SN's actual status. We omit reporting station-to-CH evaluation results here as the same trends have been observed.

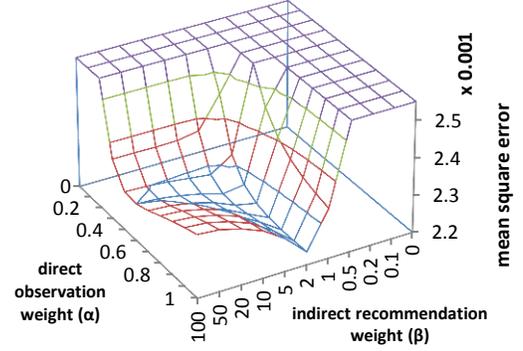


Figure 2: Effect of α and β on Accuracy of Subjective Trust Evaluation for $X=Intimacy$.

Figure 2 shows the effect of α and β on the mean square error between subjective trust obtained from Equation 2 and objective trust obtained from actual status for $X=Intimacy$. The diagrams for other trust properties exhibit a similar trend and are not shown here due to space constraints. We vary α from 0 to 1 and β from 0 to 100 to cover all possible values. We see that as α increases (using a larger α indicates that subjective trust evaluation relies more on direct observations compared with past experiences), the mean square error first decreases and then increases. Subjective trust initially approaches objective trust as more recent direct observations are used. However, there is a crossover point (e.g., $\alpha \geq 0.8$ when $\beta = 10$) after which subjective trust deviates more from objective trust because of underestimation. On the other hand, as β increases (using a larger β indicates that subjective trust evaluation relies more on indirect recommendations provided by recommenders compared with past experiences), subjective trust initially approaches objective trust, but deviates more from objective trust after a crossover point (e.g., $\beta \geq 2$ when $\alpha=0.6$) is reached. This reason is that using too much indirect recommendations in subjective trust evaluation gives malicious nodes a higher change to successfully launch good-mouthing and bad-mouthing attacks. Figure 2 shows that using $\alpha=0.8$ and $\beta=2$ yields subjective trust values very close to objective trust values in $X=Intimacy$ with the mean square error less than 0.3%.

The best α and β values intrinsically depend on the nature of each trust property as well as a given set of parameter values as those listed in Table 2 characterizing the environmental and operational conditions. We summarize the best α and β values for each trust property in Table 3. The last column "MSE" shows the mean square error between subjective trust and objective trust in trust property X . Since the trust score in individual trust property X reflects the actual trust value in property X , the combined trust score given by Equation 1 will also reflect the actual trust value given by Equation 8 (i.e., with $MSE \leq 0.9\%$ for any combination). Overall, we observe a close correlation between subjective trust evaluation and objective

trust evaluation, thus supporting our claim that subjective trust obtained as a result of executing our proposed hierarchical trust management protocol approaches true objective trust.

Table 3: Best α and β Values for Trust Property X.

Trust Property	α	β	MSE
Intimacy	0.8	2	0.3%
Honesty	0.7	1	0.9%
Energy	0.6	1	0.1%
Unselfishness	0.9	5	0.1%

VII. TRUST-BASED GEOGRAPHIC ROUTING

In this section, we apply the proposed hierarchical trust management protocol to *trust-based geographic routing* as an application. In *geographic routing*, a node disseminates a message to a maximum of L neighbors closest to the destination node (or the sink node). In *trust-based geographic routing*, node i forwards a message to a maximum of L neighbors not only closest to the destination node but also with the highest trust values $T_{ij}(t)$. We conduct a performance analysis to compare our trust-based geographic routing protocol with baseline routing protocols, namely, flooding-based [33] and traditional geographic routing. In *flooding-based routing*, a node floods a message to all its neighbors until a copy of the packet reaches the destination node. It yields the highest message delivery ratio and the lowest message delay at the expense of the highest message overhead.

Recall that for all routing protocols, the source SN first forwards a message to its CH (through multiple hops if necessary). Then, the CH forwards the message to the sink node through other CHs. Without loss of generality, we normalize the average delay for forwarding a message between two neighbor SNs to τ . The average delay between two neighbor CHs is normalized to 2τ . We collect data for delivering 1000 messages, each with a source sensor and a sink node randomly selected. We consider two cases: $L=1$ and $L=2$ for both *trust-based geographic routing* and *geographic routing*. In the comparative analysis, we vary the degree of selfish or compromised nodes from 0% to 90%. Note that 30% of compromised or selfish nodes means that 30% of nodes are compromised or selfish in the system without a fixed ratio being used for these two types of nodes. We use parameter values as listed in Table 2 for characterizing environmental and operational conditions. We also use the optimal set of (α, β) for each individual trust property as identified in Section VI (see Table 3) to ensure subjective trust is close to objective trust.

A. Best Trust Formation to Maximize Application Performance

We first identify the best way to form trust out of social and QoS trust properties (i.e., identifying weights to assign to individual trust properties) so that the performance of trust-based geographical routing is maximized. Without loss of generality and for ease of disposition, we assume that the weights assigned to social trust properties, i.e., intimacy and honesty, are the same each of $0.5 \times w_{social}$, and the weights assigned to QoS trust properties, i.e., energy and unselfishness, are the same each of $0.5 \times w_{QoS}$ with $w_{social} + w_{QoS} = 1$. Figure 3

shows the effect of w_{social} on the message delivery ratio of trust-based geographic routing with varying population percentage of compromised or selfish nodes. We observe that using solely either social trust ($w_{social}=1$) or QoS trust ($w_{social}=0$) yields a lower message delivery ratio, while considering both social and QoS trust properties helps generate a higher message delivery ratio. Figure 3 identifies that for the example WSN described in Section III characterized by a set of parameter values listed in Table 2, the maximum message delivery ratio performance is obtained when $w_{social}=0.4$ and $w_{QoS}=0.6$. Hence, this weight setting represents the best trust formation in the trust-based geographical routing application.

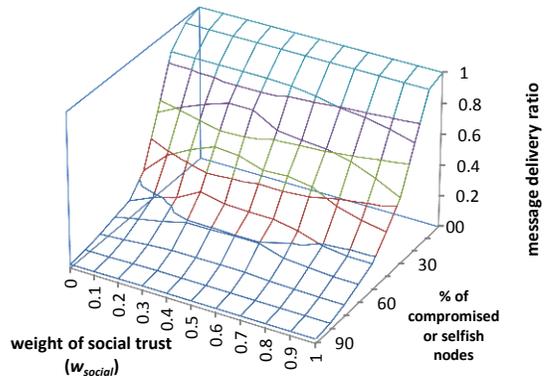


Figure 3: Effect of w_{social} on Message Delivery Ratio.

B. Dynamic Trust Management

Figure 3 illustrates the utility of dynamic trust management and application-level trust optimization for trust-based geographic routing applications, i.e., when the system senses that the hostility expressed in terms of the percentage of compromised or selfish nodes (the Y coordinate of Figure 3) is increasing, it can dynamically adjust w_{social} (the X coordinate) to optimize application performance in message delivery ratio (the Z coordinate of Figure 3).

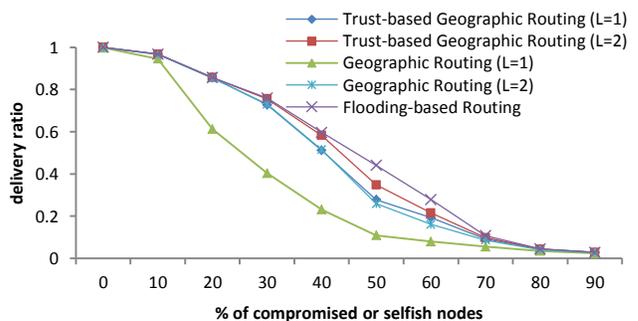


Figure 4: Message Delivery Ratio.

C. Performance Comparison

Figure 4 shows the message delivery ratio under various routing protocols. Our trust-based geographic routing protocol ($L=1$ or $L=2$) outperforms traditional geographic routing ($L=1$ or $L=2$) and approaches flooding-based routing, especially as the percentage of compromised or selfish nodes increases. The delivery ratio for all three routing protocols drops below 0.1 when the percentage of compromised or selfish nodes is higher than 80%. We observe that even the message delivery ratio of

our trust-based geographic routing without redundancy ($L=1$) is higher than that of the geographic routing with redundancy ($L=2$) when the percentage of compromised or selfish nodes is higher than 40%. We attribute this to the ability of trust-based geographic routing being able to successfully avoid forwarding messages to untrustworthy nodes based on $T_{ij}(t)$ values obtained from our hierarchical trust management protocol.

Figure 5 shows the average delay for those messages that are successfully delivered under various routing protocols for a special case in which the source SN and the sink node are at least a distance (700m) away. We create this case to ensure there are sufficient intermediate nodes on any path to reach the sink node. We first observe that the message delivery delay increases as the percentage of compromised or selfish nodes increases due to more messages being dropped by compromised or selfish nodes resided on shorter routes. Flooding-based routing has the best performance since it can always find the shortest path to reach the destination sink node through flooding. Geographic routing ($L=1$ or $L=2$) has almost the same performance with flooding-based routing due to its greedy nature for selecting nodes closest to the destination sink node for message forwarding. However, geographic routing with $L=1$ fails to deliver any message when the percentage of compromised or selfish nodes is higher than 50% because there is no short route to reach the destination node over a long distance. Trust-based geographic routing with $L=1$ has the highest delay but with $L=2$ approaches the performance of flooding-based routing and geographic routing. In general, traditional geographic routing performs better than trust-based geographic routing in message delay. This is expected because unlike traditional geographic routing, trust-based geographic routing tends to find forwarding nodes that are trustworthy but possibly not residing on the most direct path to the sink node. Consequently it incurs a higher delay compared with traditional geographic routing. However, we note that once we allow more message copies (e.g., $L=2$) to be disseminated by a node to its neighbors, trust-based geographic routing just like traditional geographic routing quickly approaches the ideal performance bound in message delay, especially as the percentage of compromised or selfish nodes increases.

Figure 6 compares message overhead in terms of the number of message copies propagated before the destination sink node receives one copy. Both geographic routing and trust-based geographic routing perform significantly better than flooding-based routing. Trust-based geographic routing incurs more message overhead than traditional geographic routing because the path selected by trust-based geographic routing is often the most trustworthy path, not necessarily the shortest path. Nevertheless, we observe that the overhead increase of trust-based geographic routing over traditional geographic routing is small compared with that of flooding-based routing over traditional geographic routing. The system thus can effectively trade off message overhead for message delivery ratio and message delay. Finally, we observe that the number of message copies propagated for all three routing protocols is close to 3 when the percentage of compromised or selfish nodes is higher than 80%. The reason is that the message can be

successfully delivered only when the source node and the sink node are close to each other. Otherwise, there is a high probability that compromised or selfish nodes reside on a long route will drop the message copies received.

Overall Figures 4-6 demonstrate that our trust-based geographic routing protocol with $L=2$ can significantly improve the delivery ratio and message delay (close to those of flooding-based routing) in the presence of compromised or selfish nodes, without sacrificing too much message overhead. Here we note that the system can effectively trade off message overhead (energy consumption) for high delivery ratio and low message delay by adjusting the level of redundancy (L). As L increases the performance of our trust-based geographic routing protocol in delivery ratio and message delay will approach that of flooding-based routing.

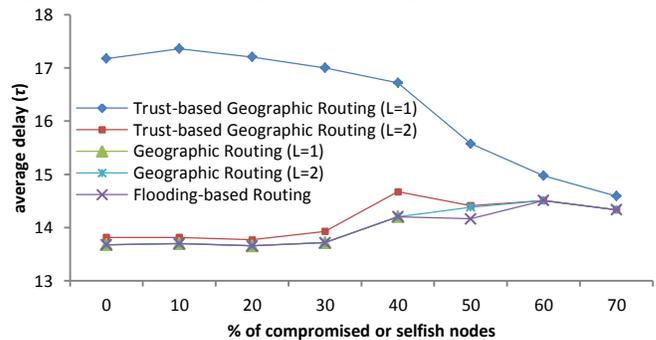


Figure 5: Message Delay with Source Sensor and Sink Node at a Distance Away.

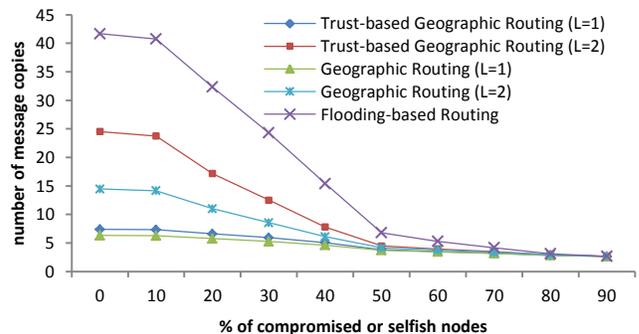


Figure 6: Message Overhead.

VIII. TRUST-BASED INTRUSION DETECTION

In this section we apply hierarchical trust management to trust-based intrusion detection as another application. We first describe the algorithm that can be used by a high-level node such as a CH (or a base station) to perform trust-based intrusion detection of the SNs (or CHs respectively) under its control. Then we develop a statistical method to assess trust-based IDS false positive and false negative probabilities.

Without loss of generality, in this section we illustrate how a CH performs trust-based intrusion detection on SNs in its cluster. A similar treatment applies to a base station performing trust-based intrusion detection on CHs in a WSN.

A. Algorithm for Trust-Based Intrusion Detection

Our trust-based IDS algorithm is based on selecting a system

minimum trust threshold, T^{th} , below which a node is considered compromised and needs to be excluded from sensor reading and routing duties. The underlying principle is that a compromised node will exhibit several social and QoS trust behaviors, i.e., low *intimacy* and low *honesty* (for social trust) as well as low *energy* and low *unselfishness* (for QoS trust), thus exposing itself as a compromised node under hierarchical trust evaluation.

A CH performs CH-to-SN trust evaluation toward node j after receiving $T_{ij}(t)$ values from all SNs in the cluster. More specifically a CH, c , when evaluating a SN, j , will compute node j 's trust value, $T_{cj}(t)$, by Equation 4. CH c will announce node j as compromised if $T_{cj}(t)$ is less than T^{th} ; otherwise, node j is not compromised.

B. Statistical Analysis

Consider that the trust value toward node j is a random variable following normal distribution commonly used for statistical analyses with mean value $\mu_j(t)$. Also consider that there are n sample values of $T_{ij}(t)$ submitted by n SNs considered trustworthy by the CH. With these n sample values, $X_j(t)$ is related to the sample mean, sample standard deviation and true mean following t-distribution with $n - 1$ degree of freedom as follows:

$$X_j(t) = \frac{\overline{T_{ij}(t)} - \mu_j(t)}{S_j(t)/\sqrt{n}} \quad (9)$$

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

$$\begin{aligned} \Theta_j(t) &= \Pr(\mu_j(t) < T^{th}) \\ &= \Pr\left(X_j(t) > \frac{\overline{T_{ij}(t)} - T^{th}}{S_j(t)/\sqrt{n}}\right) \end{aligned} \quad (10)$$

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

$$P_j^{fp}(t) = \Pr\left(X_j(t) > \frac{\overline{T_{ij}^N(t)} - T^{th}}{S_j^N(t)/\sqrt{n}}\right) \quad (11)$$

$$P_j^{fn}(t) = \Pr\left(X_j(t) \leq \frac{\overline{T_{ij}^C(t)} - T^{th}}{S_j^C(t)/\sqrt{n}}\right) \quad (12)$$

Equations 11 and 12 above give the false positive probability, $P_j^{fp}(t)$, and false negative probability, $P_j^{fn}(t)$, of our proposed trust-based intrusion detection algorithm at time t , respectively. $\overline{T_{ij}^N(t)}$ and $S_j^N(t)$ are the mean value and standard deviation of node j 's trust values reported by other nodes in the same cluster, under the condition that node j is not compromised. $\overline{T_{ij}^C(t)}$ and $S_j^C(t)$ are the mean value and standard deviation, under the condition that node j is compromised. $\overline{T_{ij}^N(t)}$ and $T_{ij}^C(t)$ can be easily obtained by applying the Bayes' theorem to the calculation of $T_{ij}(t)$.

$P_j^{fp}(t)$ and $P_j^{fn}(t)$ vary over time. The average false positive and false negative probabilities, denoted by P_j^{fp} and P_j^{fn} can be obtained by weighting on the probability of node j being compromised at time t , i.e.,

$$P_j^{fp} = \frac{\sum_{t=0}^{SL} \left(P_j^{fp}(t) (1 - P_j^C(t)) \right)}{\sum_{t=0}^{SL} (1 - P_j^C(t))} \quad (13)$$

$$P_j^{fn} = \frac{\sum_{t=0}^{SL} \left(P_j^{fn}(t) P_j^C(t) \right)}{\sum_{t=0}^{SL} P_j^C(t)} \quad (14)$$

where $P_j^C(t)$ is the probability that node j is compromised at time t which can be obtained from the SPN model output, and SL is the anticipated WNS lifetime period over which the weighted calculation is performed.

C. Best Trust Formation to Maximize Application Performance

Here we identify the best way to form trust out of social and QoS trust properties (i.e., identifying weights to assign to individual trust properties) and to assign the minimum trust threshold, T^{th} , so that the performance of trust-based intrusion detection is maximized, i.e., both false positives and false negatives are minimized. We again consider the example WSN described in Section III characterized by a set of parameter values listed in Table 2 with its lifetime $SL=150$ days.

Figure 7 shows $\max(P_{fp}, P_{fn})$ vs. T^{th} and w_{social} in this system as a result of executing our trust-based intrusion detection algorithm, where P_{fp} and P_{fn} are the time-averaged false positive and false negative probabilities as calculated from Equations 13 and 14, respectively, over all nodes in the system. We observe that as the minimum trust threshold T^{th} increases, the false negative probability P_{fn} decreases while the false positive probability P_{fp} increases. More importantly, there exists an optimal trust threshold $T^{th,opt}$ at which both false negative and false positive probabilities are minimized. As trust formation affects how trust is formed from social and QoS trust components, we also observe that $T^{th,opt}$ is sensitive to w_{social} . Figure 7 identifies that for the example WSN when $T^{th,opt} = 0.6$ and $w_{social} = 0.6$, both false positive and false negative probabilities are minimized to fall below 5%.

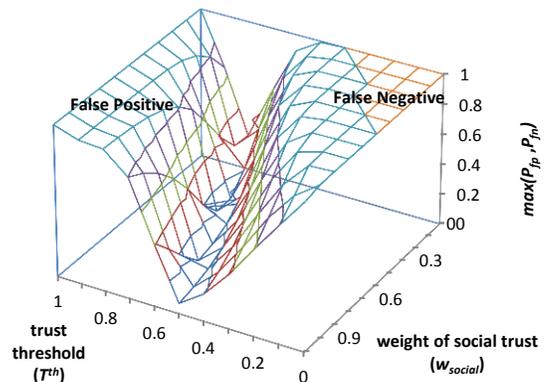


Figure 7: Effect of T^{th} and w_{social} on $\max(P_{fp}, P_{fn})$.

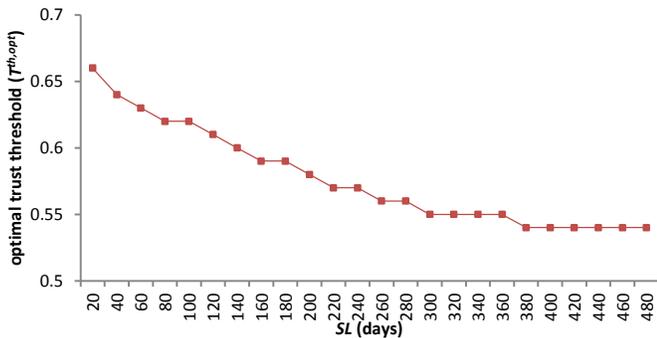


Figure 8: Optimal Trust Threshold vs. System Lifetime.

D. Dynamic Trust Management

Figure 7 is for the case in which the expected system lifetime SL is 150 days of operations. Figure 8 shows the optimal trust threshold $T^{th,opt}$ as SL varies. Here, the value of w_{social} is fixed to 0.6 to isolate its effect. For a WSN with a prolonged operation, each SL value represents a time point characterized by a distinct hostility level such as the percentage of compromised and selfish nodes. We observe that as SL increases, the value of $T^{th,opt}$ at which the false alarm probability is minimized decreases. The reason is that a node's trust value decreases over time due to energy depletion even if the node is not compromised. The system sensing hostility change at runtime can apply the best w_{social} and $T^{th,opt}$ setting identified from static analysis to optimize application performance in false alarm probability.

E. Performance Comparison

We perform a comparative performance analysis of our trust-based intrusion detection algorithm with two anomaly detection schemes, namely, weighted summation [18] and data clustering [19]. We use the ROC (Receiver Operating Characteristic) curve [19] as the performance metric since both false negative probability (P_{fn}) and false positive probability (P_{fp}) are critical measures and ROC objectively reflects the sensitivity of detection probability (i.e., $1 - P_{fn}$) as the false positive probability varies.

The first baseline anomaly detection scheme is weighted summation-based IDS [18]. In this approach, each SN has a weight associated with it and this weight changes dynamically, reflecting the trustworthiness of the SN's output relative to the average output out of all SNs. We use the trust recommendation value from each SN toward a particular SN, say, SN_i , as the SN's output. The average trust recommendation value is obtained by a summation of the trust recommendation values weighted by the respective weights from all SNs except SN_i . If the trust recommendation value from a SN deviates too much from the average value, the weight value associated with that SN decreases by θ (weight penalty); otherwise the weight value remains the same. The weight value is updated dynamically until it falls below a weight threshold (w_t), in which case the corresponding SN is reported as malicious. The weight penalty (θ) and weight threshold (w_t) largely determine the false positive probability. We vary θ and w_t over the range of [0, 1] to

obtain the detection probability as the false positive probability varies.

The second baseline anomaly detection scheme is fixed width data clustering-based IDS [19]. In this approach, the maximum radius of a cluster (c_w) is defined and a data point is put into a cluster if the distance between the centroid of the cluster and this data point is smaller than c_w ; otherwise this data point makes a new cluster. Data points that exhibit dissimilarity with others will tend to cluster into a small cluster or standalone by themselves. These lone data points are reported as malicious. To apply fixed width data clustering-based IDS, we use trust values of SNs as collected by a CH as data points for clustering. As the maximum radius of a cluster c_w affects the false positive and negative probabilities, we vary c_w over the range of [0, 0.2] to collect the performance results.

In our trust-based intrusion detection algorithm, the false positive and negative probabilities essentially depend on the minimum trust threshold (T^{th}) and the weight of social trust (w_{social}). We vary these two parameters over the range of [0, 1] to collect the performance results.

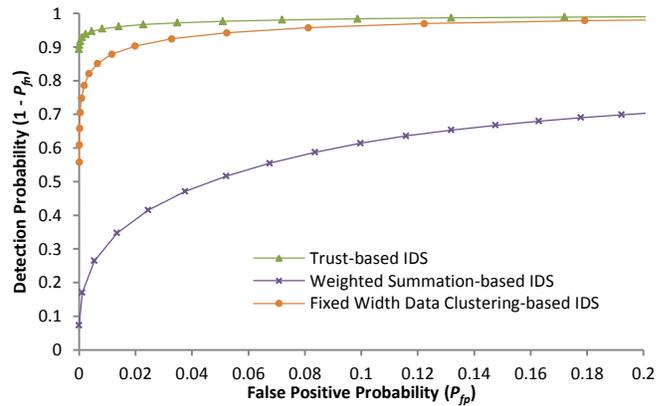


Figure 9: Performance Comparison of IDS Schemes in Detection Probability vs. False Positive Probability.

In Figure 9 we compare the ROC curves of our trust-based IDS algorithm against those by weighted summation-based IDS and fixed width data cluster-based IDS for $SL=240$ days. The results presented are the best results of all three IDS schemes by fine-tuning the design parameters as described above under the same network environment characterized by Table 2.

We observe from Figure 9 that as a design tradeoff, as the false positive probability increases, the detection probability increases for all IDS schemes. We observe that our trust-based IDS algorithm outperforms both weighted summation-based IDS and fixed width data clustering-based IDS, especially when the false positive probability is limited to 5% which is considered desirable in intrusion detection. The strength of our trust-based IDS algorithm is especially pronounced when the false positive probability approaches zero. This is very desirable since our trust-based IDS algorithm can still maintain a high detection probability (>90%) when the false positive probability is close to zero at which the detection probability of anomaly detection-based IDS schemes drops sharply.

IX. CONCLUSION

In this paper, we proposed a hierarchical dynamic trust management protocol for cluster-based wireless sensor networks, considering two aspects of trustworthiness, namely, social trust and QoS trust. We developed a probability model utilizing stochastic Petri nets techniques to analyze the protocol performance, and validated subjective trust against objective trust obtained based on ground truth node status. We demonstrated the feasibility of dynamic hierarchical trust management and application-level trust optimization design concepts with trust-based geographic routing and trust-based IDS applications, by identifying the best way to form trust as well as use trust out of individual social and QoS trust properties at runtime to optimize application performance. The results indicated that our trust-based geographic routing protocol performs close to the ideal performance of flooding-based routing in delivery ratio and message delay without sacrificing much in message overhead compared with traditional geographic routing protocols which do not use trust. Our trust-based IDS algorithm outperforms traditional anomaly-based IDS techniques in the detection probability while maintaining sufficiently low false positives.

There are several future research directions, including (a) devising and validating a decentralized trust management scheme for autonomous WSNs without base stations; (b) investigating the impact of the cluster size and the trust update interval to the protocol performance and lifetime of a given WSN; and (c) investigating the feasibility of applying hierarchical trust management to more dynamic networks such as mobile WSNs, mobile cyber physical systems, or MANETs.

REFERENCES

- [1] 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, 2011, pp. 562-583.
- [2] 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.
- [3] R. A. Sahner, K. Trivedi, and A. Puliafito, *Performance and Reliability Analysis of Computer Systems*: Kluwer Academic Publishers, 1996.
- [4] B. Karp, and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *ACM International Conference on Mobile Computing and Networking*, Boston, USA, August 2002, pp. 243-254.
- [5] Y. J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic Routing Made Practical," in *USENIX/ACM Symposium on Networked System Design and Implementation*, Boston, USA, May 2005, pp. 217-230.
- [6] E. Aivaloglou, and S. Gritzalis, "Hybrid Trust and Reputation Management for Sensor Networks," *Wireless Networks*, vol. 16, no. 5, July 2010, pp. 1493-1510.
- [7] S. Ganerwal, L. K. Balzano, and M. B. Srivastava, "Reputation-Based Framework for High Integrity Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 3, May 2008, pp. 1-37.
- [8] K. Liu, N. Abu-ghazaleh, and K. D. Kang, "Location Verification and Trust Management for Resilient Geographic Routing," *Journal of Parallel Distributed Computing*, vol. 67, no. 2, Feb. 2007, pp. 215-228.
- [9] L. Moraru, P. Leone, S. Nikolettseas, and J. D. P. Rolim, "Near Optimal Geographic Routing with Obstacle Avoidance in Wireless Sensor Networks by Fast-Converging Trust-Based Algorithms," in *ACM Workshop on QoS and Security for Wireless and Mobile Networks*, Chania, Crete Island, Greece, 2007, pp. 31-38.
- [10] R. A. Shaikh, et al., "Group-Based Trust Management Scheme for Clustered Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 11, Nov. 2009, pp. 1698-1712.
- [11] Z. Yao, D. Kim, and Y. Doh, "PLUS: Parameterized and Localized Trust Management Scheme for Sensor Networks Security," in *IEEE International Conference on Mobile Adhoc and Sensor Systems*, Vancouver, BC, Oct. 2006, pp. 437-446.
- [12] L. Capra, and M. Musolesi, "Autonomic Trust Prediction for Pervasive Systems," in *International Conference on Advanced Information Networking and Applications*, April 2006, pp. 1-5.
- [13] L. Capra, "Engineering Human Trust in Mobile System Collaborations," in *12th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 2004.
- [14] J. Zhang, et al., "A Trust Management Architecture for Hierarchical Wireless Sensor Networks," in *IEEE Conference on Local Computer Networks*, Denver, Colorado, Oct. 2010, pp. 264-267.
- [15] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Anomaly Detection in Wireless Sensor Networks," *IEEE Wireless Communications*, vol. 15, no. 4, August 2008, pp. 34-40.
- [16] V. Bhuse, and A. Gupta, "Anomaly Intrusion Detection in Wireless Sensor Network," *J. of High Speed Networks*, vol. 15, no. 1, Jan. 2006, pp. 33-51.
- [17] A. da Silva, et al., "Decentralized Intrusion Detection in Wireless Sensor Networks," in *ACM International Workshop on Quality of Service and Security in Wireless and Mobile Networks*, Montreal, Quebec, Canada, Oct. 2005, pp. 16-23.
- [18] H. Hu, et al., "Weighted Trust Evaluation-Based Malicious Node Detection for Wireless Sensor Networks," *International Journal of Information and Computer Security*, vol. 3, no. 2, 2009, pp. 132-149.
- [19] C. E. Loo, M. Y. Ng, C. Leckie, and M. Palaniswami, "Intrusion Detection for Routing Attacks in Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 2, no. 4, 2006, pp. 313-332.
- [20] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, "Quarter Sphere Based Distributed Anomaly Detection in Wireless Sensor Networks," in *IEEE International Conference on Communications*, Glasgow, U.K., June 2007, pp. 3864-3869.
- [21] F. Wang, C. Huang, J. Zhao, and C. Rong, "IDMTM: A Novel Intrusion Detection Mechanism Based on Trust Model for Ad Hoc Networks," in *International Conference on Advanced Information Networking and Applications*, March 2008, pp. 978-984.
- [22] P. Ebinger, and N. Bissmeyer, "TEREC: Trust Evaluation and Reputation Exchange for Cooperative Intrusion Detection in MANETs," in *Communication Networks and Services Research Conference*, May 2009, pp. 378-385.
- [23] G. Theodorakopoulos, and J. S. Baras, "On Trust Models and Trust Evaluation Metrics for Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, Feb. 2006, pp. 318-328.
- [24] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor Network Security: A Survey," *IEEE Communication Surveys & Tutorials*, vol. 11, no. 2, 2009, pp. 52-73.
- [25] F. Bao, I. R. Chen, M. Chang, and J. H. Cho, "Hierarchical Trust Management for Wireless Sensor Networks and Its Application to Trust-Based Routing," in *ACM Symposium on Applied Computing*, TaiChung, Taiwan, March 2011, pp. 1732-1738.
- [26] F. Bao, I. R. Chen, M. Chang, and J. H. Cho, "Trust-Based Intrusion Detection in Wireless Sensor Networks," in *IEEE International Conference on Communications*, Kyoto, Japan, June 2011, pp. 1-6.
- [27] O. Younis, and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-hoc Sensor Networks," *IEEE Trans. on Mobile Computing*, vol. 3, no. 3, Oct.-Dec. 2004, pp. 366-379.
- [28] P. B. Velloso, et al., "Trust Management in Mobile Ad Hoc Networks Using a Scalable Maturity-Based Model," *IEEE Transactions on Network and Service Management*, vol. 7, no. 3, Sept. 2010, pp. 172-185.
- [29] M. S. Islam, R. H. Khan, and D. M. Bappy, "A Hierarchical Intrusion Detection System in Wireless Sensor Networks," *Computer Science and Network Security*, vol. 10, no. 8, August 2010, pp. 21-26.
- [30] C. Budianu, S. Ben-David, and L. Tong, "Estimation of the Number of Operating Sensors in Large-Scale Sensor Networks with Mobile Access," *IEEE Trans. on Signal Processing*, vol. 54, no. 5, May 2006, pp. 1703-1715.
- [31] R. A. F. Mini, A. A. F. Loureiro, and B. Nath, "The Distinctive Design Characteristic of A Wireless Sensor Network: the Energy Map," *Computer Communications*, vol. 27, no. 10, June 2004, pp. 935-945.
- [32] Y. J. Zhao, R. Govindan, and D. Estrin, "Residual Energy Scan for Monitoring Sensor Networks," in *IEEE Wireless Communication and Networking Conference*, Orlando, USA, March 2002, pp. 356-362.
- [33] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case," *IEEE/ACM Trans. on Networking*, vol. 16, no. 1, Feb. 2008, pp. 77-90.