Coping with Node Misbehaviors in Ad Hoc Networks: A Multi-Dimensional Trust Management Approach

Wenjia Li, Anupam Joshi (*IEEE Senior Member*), and Tim Finin
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County (UMBC)
Baltimore, Maryland 21250
{wenjia1, joshi, finin}@cs.umbc.edu

***Abstract*—Nodes in Mobile Ad hoc Networks (MANETs) are required to relay data packets to enable communication between other nodes that are not in radio range with each other. However, whether for selfish or malicious reasons, a node may fail to cooperate during the network operations or even attempt to disturb them, both of which have been recognized as misbehaviors. Various trust management schemes have been studied to assess the behaviors of nodes so as to detect and mitigate node misbehaviors in MANETs. Most of existing schemes model a node's trustworthiness along a single dimension, combining all of the available evidence to calculate a single, scalar trust metric. A single measure, however, may not be expressive enough to adequately describe a node's trustworthiness in many scenarios. In this paper, we describe a multi-dimensional framework to evaluate the trustworthiness of MANET node from multiple perspectives. Our scheme evaluates trustworthiness from three perspectives: collaboration trust, behavioral trust, and reference trust. Different types of observations are used to independently derive values for these three trust dimensions. We present simulation results that illustrate the effectiveness of the proposed scheme in several scenarios.**

***Index Terms*—Mobile Ad hoc Network; Misbehavior; Security; Multi-dimensional trust**

I. Introduction

A Mobile Ad hoc Network (MANET), as is implied by its name, is normally composed of a dynamic set of cooperative nodes that are willing to relay packets for other nodes due to the lack of any pre-deployed network infrastructure. The nature of the mobile nodes in MANET makes them extremely vulnerable to a variety of security threats because they usually own low computational resource as well as short radio transmission range due to the limited battery power they carry, and they might be moving constantly. For instance, owing to the open transmission medium as well as the lack of authentication infrastructure, wireless links in MANET are more inclined to both passive eavesdropping and active tampering when compared to the traditional wired network. Therefore, security is one of the most important challenges for MANET.

Node misbehavior is such a category of security threat for Mobile Ad hoc Networks (MANETs). In general, misbehaviors can be conducted at every layer in MANETs, such as malicious flooding of the Request-To-Send (RTS) frames in the MAC layer, dropping, modification, and misroute to the packets in the network layer, and deliberate propagation of fake observations regarding the behaviors of other nodes in the application layer. Moreover, node misbehaviors may range from lack of cooperation to active attacks aiming at Denial-of-Service (DoS) and subversion of traffic. For example, because of the limited resources (such as battery power and bandwidth, etc) that each node can possibly possess, a *selfish* node may choose not to cooperate with other nodes so as to preserve its own resources [1]. In other words, when a *selfish* node is requested to forward some data packets for other nodes, it might drop a part or all of the incoming packets. By this means, it can save the battery power and transmit some extra packets for the sake of itself. On the other hand, some *malicious* nodes aim to disturb the network services, and they may intentionally drop, modify or misroute packets while it is not a priority for them to save battery lives [2]. Regardless of the intents by which the node misbehaviors are induced, they are obviously harmful to a currently healthy MANET.

To address the security vulnerabilities caused by various node misbehaviors in Mobile Ad hoc Networks (MANETs), numerous security solutions have been proposed to detect and mitigate those misbehaviors from distinctive perspectives, such as the mechanisms discussed accordingly in [3], [4], and [5]. Because it is quite beneficial to assess a node's behaviors and determine if it is trustworthy in terms of how cooperative it is, trust management mechanism has become a power tool to cope with node misbehaviors. A variety of trust management mechanisms have been studied during the past decades, such as the mechanisms discussed in [5], [6], and [7]. Most of these trust management mechanisms model the trust of a node in one dimension, i.e., all available evidence and observations are utilized to calculate a single, scalar trust metric for each node. However, a single trust metric may not be expressive enough to adequately describe whether a node is trustworthy or not in many complicated scenarios. Figure 1 demonstrates such a scenario in which a single trust measure is not expressive enough.

From Figure 1, we can find that in the first step, the observer

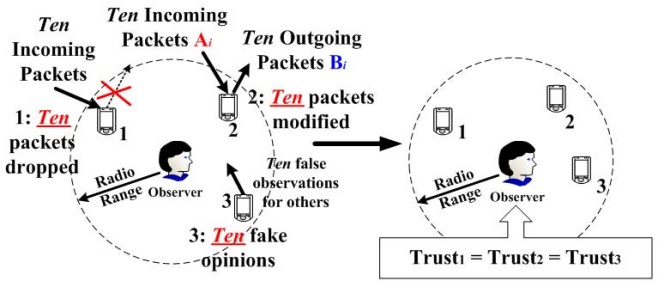


Fig. 1. A Scenario Where A Single Trust is NOT Expressive Enough

collects and records various misbehaviors that are conducted by node 1, 2, and 3. The observation results illustrate that node 1, 2, and 3 have dropped packets, modified packets, and propagated to other nodes some false observations at an amount of 10, respectively. Suppose that these three misbehaviors are punished at a same rate when the trustworthiness of each node is evaluated. Then, the observer may draw a conclusion that the trustworthiness of all these three nodes are identical. As a result, the observer will equally treat node 1 and node 3 when it needs to determine which node to forward packets as well as which node it should believe when exchanging opinions. However, it is quite apparent that the trustworthiness of node 1 and node 3 is not equivalent to each other when it comes to both packet forwarding and observation exchanging. Hence, we can safely conclude that it is neither accurate nor effective to merely use one single scalar when the trustworthiness of a node is assessed.

In this paper, a *multi*-dimensional trust management framework is proposed to better evaluate the trustworthiness of nodes in MANETs. Compared to the traditional *single*-dimensional trust management mechanisms, the trustworthiness of a node is judged from different *perspectives* (i.e., *dimensions*), and each dimension of the trustworthiness is derived from various sets of misbehaviors according to the nature of those misbehaviors. In our scheme, each node in MANETs first observes and collect the abnormal behaviors that its neighboring nodes have conducted, and this observation process is similar to existing mechanisms such as [8], [9], [10]. Unlike the majority of these mechanisms, every node locally detects the misbehaviors and infer the trustworthiness (in terms of different dimensions) of its neighbors from its own observations in the second step. Next, the observations are exchanged amongst the neighboring nodes, and the local views of misbehavior as well as trust will be updated accordingly only when some brand new observations are offered by a trustworthy neighbor. In case that there is any update in the local views, the updated observations will be further broadcast to the immediate neighbors. The observation exchange process will last until there is not any local view update for all the nodes.

The major contribution of this paper is to explore how misbehaviors can be correctly identified and trustworthiness be properly evaluated in existence of these unreliable observations. The key novel components of the proposed approach are: (1) the multi-dimensional trust management framework in which the notion of trustworthiness is further classified into several attributes (i.e., dimensions) so that each attribute is able to precisely indicate whether or not a node is trustworthy in terms of one specific behavior that it should conduct, such as cooperation, well-behaving, and honest; and (2) an adaptive trust evolution model by which each dimension of the trustworthiness can be adjusted according to the features of the misbehavior to which the dimension is related, such as severity of the outcome, frequency of occurrence, and context in which the misbehavior occurs. The multi-dimensional trust management framework: (i) allows nodes to derive accurate assessments to other nodes in MANETs; (ii) detects and mitigates node misbehaviors from both reliable and unreliable observations; and (iii) is resilient to fake opinions spread by misbehaving nodes.

The remainder of the paper is organized as follows. Section 2 briefly reviews the literature on misbehavior detection as well as trust management for MANETs. In Section 3, the proposed multi-dimensional trust management framework is discussed in more details. Section 4 presents the simulation results that we have obtained to validate our proposed approach. Finally, we draw the conclusion and point out several future directions in Section 5.

II. Related Work

In recent years, there has been significant research interest in the topics of misbehavior detection as well as trust management for ad hoc networks. Hence, the related work for these two research topics will be presented separately in this section.

A. Misbehavior Detection for Ad hoc Networks

When it comes to the discussion of misbehavior detection, we should first clearly understand the term *misbehavior* itself. Note that the term *misbehavior* generally refers to abnormal behavior that deviates from the set of behaviors that each node is supposed to conduct in MANETs [11]. According to [12], there are four types of misbehaviors in ad hoc networks, namely failed node behaviors, badly failed node behaviors, selfish attacks, and malicious attacks. These four types of node misbehaviors are classified with respect to the node's intent and action. More specifically, selfish attacks are intentional passive misbehaviors, where nodes choose not to fully participate in the packet forwarding functionality to conserve their resources, such as battery power; malicious attacks are intentional active misbehaviors, where the malicious node aims to purposely interrupt network operations. The existence of selfishness and malicious behaviors has remarkably motivated research in the area of misbehavior detection for mobile ad hoc networks.

Intrusion Detection System (IDS) is normally regarded as an important solution for detecting various node misbehaviors in MANETs. Several approaches have been proposed to build IDS probes on each individual peer due to the lack of a fixed

infrastructure, such as [3], [13], [14]. In these approaches, there is one IDS probe installed on each node, and each IDS probe is assumed to be always monitoring the network traffic, which is obviously not energy efficient given the limited battery power that each node has in MANETs. In contrast, Huang et al. [8] proposed a cooperative intrusion detection framework in which clusters are formed and the nodes in each cluster will fulfill the intrusion detection task in turn. This cluster-based approach can noticeably reduce the power consumption for each node.

Routing misbehaviors are another major security threats that have been extensively studied in ad hoc networks. In addition to externally intruding into MANETs, an adversary may also choose to compromise some nodes in ad hoc networks, and make use of them to disturb the routing services so as to make part of or the entire network unreachable. Marti et al. [4] introduced two related techniques, namely *watchdog* and *pathrater*, to detect and isolate misbehaving nodes, which are nodes that do not forward packets. There are also some other solutions that aim to cope with various routing misbehaviors [15], [16], [17].

We have also made some efforts to address the problem of misbehavior detection in ad hoc networks [18], [19], [20]. In our initial work [18], we have done a preliminary study where outlier detection method is adopted to identify node misbehaviors. The work in [19] extends the idea in that both weighted voting and the Dempster-Shafer Theory of evidence (DST) are used to combine multiple pieces of evidences from different observations in order to detect the node misbehaviors in a more accurate manner. In our latest work [20], policy as well as context information have been utilized to reveal the difference between the truly malicious nodes and the faulty ones, both of which may be treated as misbehaving nodes with no difference in most of existing misbehavior detection mechanisms.

B. Trust Establishment and Management in Ad hoc Networks

The main purpose of trust management is to assess various behaviors of other nodes and build a reputation for each node based on the behavior assessment. The reputation can be utilized to decide trustworthiness for other nodes, make choices on which nodes to cooperate with, and even take action to punish an untrustworthy node if necessary.

In general, the trust management system usually relies on two sorts of observations to evaluate the node behaviors. The first kind of observation is named as *first-hand* observation, or in other words, direct observation [21]. First-hand observation is the observation that is directly made by the node itself, and the first-hand observation can be collected either passively or actively. If a node promiscuously observes its neighbors' actions, the local information is collected passively. In contrast, the reputation management system can also rely on some explicit evidences to assess the neighbor behaviors, such as an acknowledge packet during the route discovery process. The other kind of observation is called *second-hand* observation or indirect observation. Second-hand observation is generally obtained by exchanging first-hand observations with other nodes in the network. The main disadvantages of second-hand observations are related to overhead, false report and collusion [22], [23].

In [5], Buchegger et al. proposed a protocol, namely CONFIDANT (Cooperation Of Nodes, Fairness In Dynamic Ad-hoc NeTworks), to encourage the node cooperation and punish misbehaving nodes. CONFIDANT has four components in each node: a Monitor, a Reputation System, a Trust Manager, and a Path Manager. The Monitor is used to observe and identify abnormal routing behaviors. The Reputation System calculates the reputation for each node in accordance with its observed behaviors. The Trust Manager exchanges alerts with other trust managers regarding node misbehaviors. The Path Manager maintains path rankings, and properly responses to various routing messages. A possible drawback of CONFIDANT is that an attacker may intentionally spread false alerts to other nodes that a node is misbehaving while it is actually a well-behaved node. Therefore, it is important for a node in CONFIDANT to validate an alert it receives before it accepts the alert.

Michiardi et al. [1] presented a mechanism with the name CORE to identify selfish nodes, and then compel them to cooperate in the following routing activities. Similar to CONFIDANT, CORE uses both a surveillance system and a reputation system to observe and evaluate node behaviors. Nevertheless, while CONFIDANT allows nodes exchange both positive and negative observations of their neighbors, only positive observations are exchanged amongst the nodes in CORE. In this way, malicious nodes cannot spread fake charges to frame the well-behaved nodes, and consequently avoid denial of service attacks toward the well-behaved nodes. The reputation system maintains reputations for each node, and the reputations are adjusted upon receiving of new evidences. Since selfish nodes reject to cooperate in some cases, their reputations are lower than other nodes. To encourage node cooperation and punish selfishness, if a node with low reputation sends a routing request, then the request will be ignored and the bad reputation node cannot use the network.

Patwardhan et al. [24] studied an approach in which the reputation of a node is determined by data validation. In this approach, a few nodes, which are named as Anchor nodes here, are assumed to be pre-authenticated, and thus the data they provide are regarded as trustworthy. Data can be validated by either agreement among peers or direct communication with an anchor node. Malicious node can be identified if the data they present is invalidated by the validation algorithm.

III. General Framework

In this section, we present the proposed multi-dimensional trust management framework in details. The goal of the framework is to identify and mitigate misbehaviors by means of behavior observation and trustworthiness assessment to the nodes.

A. Preliminaries

The term *node* is defined as a system entity in MANETs that owns a *tiny* processor that has limited computational capability as well as a wireless Network Interface Card (NIC) with a bounded radio transmission range. Moreover, we also assume that each node is capable of observing the behaviors of other nodes within its radio transmission range, and exchanging these observations with other nodes in its radio transmission range. Denote a *neighbor* of a node A as a node that resides within A's radio transmission range. The type of abnormal behaviors that each node observes can be defined by the nodes themselves as long as all the nodes observe the same set of abnormal behaviors.

While a node observes the abnormal behaviors that its neighbors conduct, it also keeps track of the total amount of incoming packets it has observed for each neighbor. When a node needs to summarize its observation and thereby form its local view of misbehaving nodes, it will calculate the rate of abnormal behaviors over the overall behaviors it has observed for the node. For instance, if all the nodes choose to observe the behaviors of packet drop, modification and misroute, then *packet drop rate* (PDR), *packet modification rate* (PMOR) and *packet misroute rate* (PMIR) can be defined as follows, respectively.

$$PDR = \frac{Number\ of\ Packet\ Dropped}{Total\ Number\ of\ Incoming\ Packets}$$

$$PMOR = \frac{Number\ of\ Packet\ Modified}{Total\ Number\ of\ Incoming\ Packets}$$

$$PMIR = \frac{Number\ of\ Packet\ Misrouted}{Total\ Number\ of\ Incoming\ Packets}$$

We define the *trustworthiness* of a node N_k as a vector $\Theta_k = (\theta_k^{(1)},\ \theta_k^{(2)},\ \cdots,\ \theta_k^{(n)})$, in which $\theta_k^{(i)}$ stands for the i-th dimension of the trustworthiness for the node N_k. Each dimension of the trustworthiness $\theta_k^{(i)}$ corresponds to one or a certain category of behavior(s) $B_k^{(i)}$ (such as packet forwarding or true opinion spreading), and $\theta_k^{(i)}$ can properly reflect the probability with which the node will conduct $B_k^{(i)}$ in an appropriate manner. $\theta_k^{(i)}$ can be assigned any real value in the range of $[0, 1]$, i.e., $\forall i \in \{1,\ 2,\ \cdots,\ n\}$, $\theta_k^{(i)} \in [0, 1]$. The higher the value of $\theta_k^{(i)}$, the node N_k is more likely to conduct $B_k^{(i)}$ in a proper manner.

Each dimension of the trustworthiness $\theta_k^{(i)}$ for the node N_k is defined as a function of the misbehaviors $M_k^{(i)}$ that are related to $B_k^{(i)}$ and have been observed by the neighbors of the node N_k. Different dimensions of the trustworthiness may correspond to different functions, and the selection of different functions should coincide with the basic features of $M_k^{(i)}$, such as severity of the outcome, occurrence frequency, and context in which they occur.

B. Framework Overview

In the trust management framework, there are two major functional modules, namely *Trust Management* and *Neighbor Behavior Assessment*. These two modules can support the following functionalities: *Neighbor Behavior Observation*, *Local View Formation and Update*, *View Combination*, *Trust Establishment and Evolution*, and *Local Observation Exchange*. Figure 2 illustrates our framework.

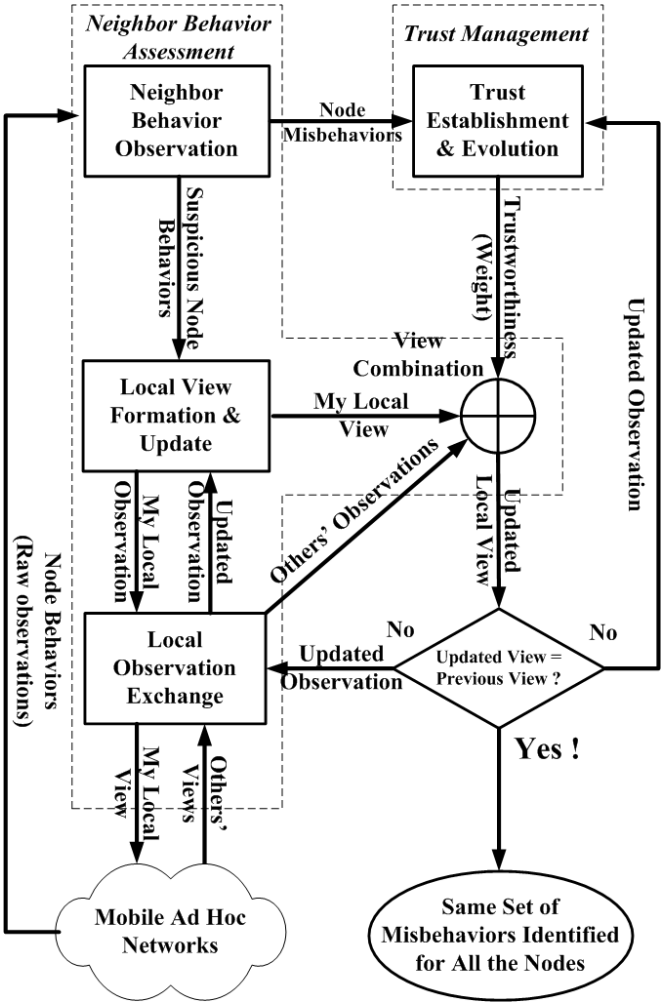


Fig. 2. Multi Dimensional Trust Management Framework

Prior to the initial local view formation, each node observes and records the behaviors of their neighbors, and also keeps track of the total number of incoming packets that each of their neighbors has received. Based on the observation, the initial local view of misbehaviors is formed. At the same time, the initial trustworthiness is established for each neighbor based on their behaviors.

Each node then exchanges its local observation with its neighbors. If a node finds that there is any *new* observation that it has never seen before from a trustworthy neighbor, it will integrate the new observation to its current view of misbehaviors, and it will broadcast the updated view to its neighbor. Note that the trustworthiness is also updated in accordance to the updated view of misbehaviors.

Once all the nodes notice that they are not receiving any new observation from their neighbors, the process halts and all the

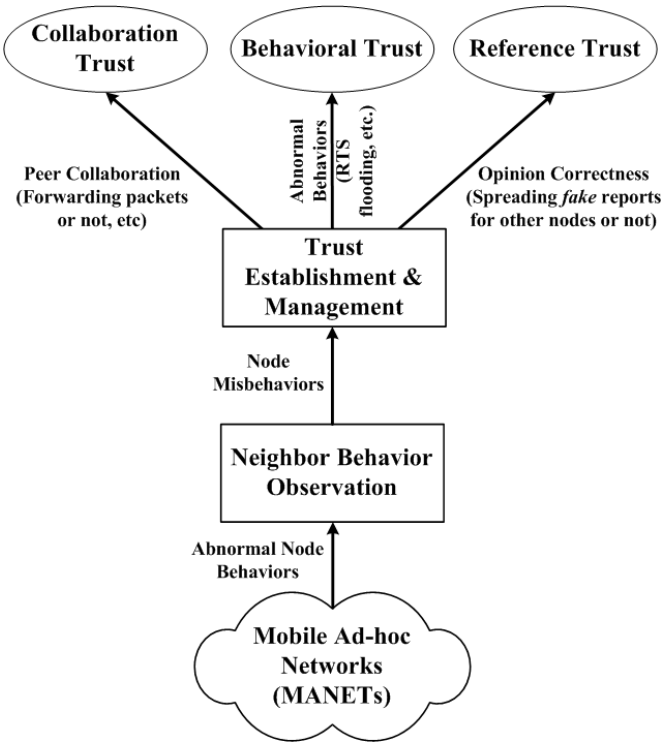


Fig. 3. The Three Dimensions of Trustworthiness

nodes converge to a unique global view of misbehaviors. Due to node mobility as well as changing network topology, the status of nodes and network changes over time. Hence, the global view of misbehaviors may become out-of-date because it can only indicate the status of nodes at the time when it was derived. To address this problem, we can periodically restart the process in order to keep the global view up-to-date. The repeat interval can be determined by both the availability of resources (such as bandwidth, battery power, etc.) and the levels of node mobility as well as topology change.

C. Multi-dimensional Trust Management

In the multi-dimensional trust management framework, the trustworthiness of a node N_k is currently assessed in three dimensions, i.e., $\Theta_k = (\theta_k^{(1)},\ \theta_k^{(2)},\ \theta_k^{(3)})$. The three dimensions $\theta_k^{(1)}$, $\theta_k^{(2)}$, and $\theta_k^{(3)}$ are called *Collaboration Trust* (CT), *Behavioral Trust* (BT), and *Reference Trust* (RT), respectively. The three dimensions of trustworthiness are demonstrated in Figure 3.

From Figure 3 we may find that CT ($\theta_k^{(1)}$) is determined by how collaborative a node N_k would be when it is asked to participate in some network activities such as route discovery and packet forwarding. BT ($\theta_k^{(2)}$) is derived by the amount of abnormal behavior that N_k has conducted, including packet modification, packet misroute or RTS flooding attack. RT ($\theta_k^{(3)}$) is generally computed based on the correctness of the observation that N_k spreads. For instance, if N_k has been witnessed repeatedly sending fake observations to its neighbors, $\theta_k^{(3)}$ should be assigned a rather low value. In this way, other nodes can properly interpret or even ignore the observations offered by N_k because $\theta_k^{(3)}$ is used as the weight for N_k when those observations are integrated to the local views of others.

Note that different categories of misbehaviors may occur in different contexts. Moreover, the consequences that these misbehaviors lead to can range significantly from loss of one packet to a benign node being framed by fake opinions and consequently trapped into denial of service. However, most of the existing trust management schemes have hardly taken these factors into consideration, and they generally punish all the misbehaviors on a uniform scale when the trustworthiness is derived. To better take the features of misbehaviors into account, we have developed an *adaptive* trustworthiness evolution model for different dimensions of trustworthiness, or even for the same dimension in different contexts.

Let us take the three dimensions of trustworthiness that we define as an example. Given that packet dropping may be caused by both malicious intent and environmental factors such as overflown buffer and exhausted battery, *Collaboration Trust* ($\theta_k^{(1)}$) should be reduced at a lower rate when compared to *Behavioral Trust* ($\theta_k^{(2)}$) because both packet modification and flooding of RTS frames can never be owed to environmental factors. Similarly, it is really harmful to spread fake observations in MANETs because the fake observations can cause massive chaos when the nodes attempt to tell trustworthy neighbors from untrustworthy ones from their behaviors. As a result, *Reference Trust* ($\theta_k^{(3)}$) should decrease at the highest rate when compared to both CT and BT. Based on these arguments, we may utilize *logarithmic model*, *linear model*, and *exponential model* for $\theta_k^{(1)}$, $\theta_k^{(2)}$, and $\theta_k^{(3)}$, respectively. In other words, the following formulas should hold for the trustworthiness of the node N_k.

$$\theta_k^{(1)} \propto (a_1 * \log M_k^{(1)} + b_1)\ ,\ a_1, b_1 \in \mathbb{Q}$$

$$\theta_k^{(2)} \propto (a_2 * M_k^{(2)} + b_2)\ ,\ a_2, b_2 \in \mathbb{Q}$$

$$\theta_k^{(3)} \propto (a_3 * c_3^{M_k^{(3)}} + b_3)\ ,\ a_3, b_3, c_3 \in \mathbb{Q}$$

In our simulation, we have tried different sets of functions that satisfy these conditions. As an example, we find out that the following set of functions is feasible in practice to derive the three dimensions of trustworthiness. Figure 4 illustrates these functions for the three dimensions of trustworthiness.

$$\theta_k^{(1)} = \sqrt{\log_2 (2 - M_k^{(1)})}$$

$$\theta_k^{(2)} = 1 - M_k^{(2)}$$

$$\theta_k^{(3)} = \exp(-5 * M_k^{(3)})$$

Not only can the trust evolution model differs for different dimensions of trustworthiness, it can also alter for the same dimension in different context. For example, because packet dropping may be caused by both malicious intent and environmental factors, we can collect the context in which packet dropping occurs. If we infer from the context that it is caused

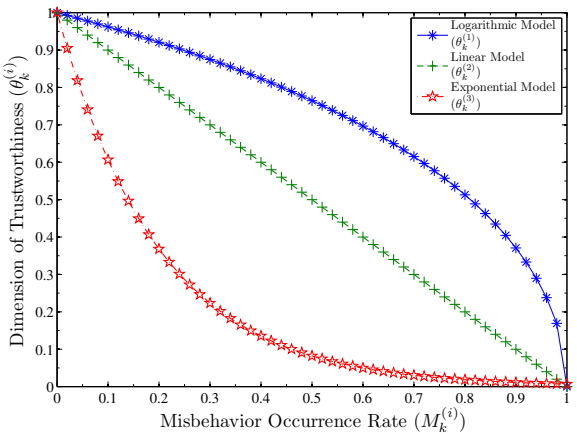


Fig. 4. Various Trustworthiness Evolution Models

by environmental factors, then we can use *logarithmic model* for $\theta_k^{(1)}$. In contrast, if we decide that the packet dropping is the outcome of malicious intent, then we may use *linear model* for $\theta_k^{(1)}$ to speed up the punishment.

D. View Combination

View combination is one of the most important functionalities in our proposed framework. Because some of the incoming observations are not reliable, it is essential to find a view combination technique to properly fuse together multiple pieces of views in presence of both trustworthy and untrustworthy observations.

As we have discussed in [19], Dempster-Shafer theory of evidence (DST) [25] is an appropriate technique to fuse together multiple piece of observations even if some of them might not be accurate. In DST, probability is replaced by an uncertainty interval bounded by belief (bel) and plausibility (pls). Belief is the lower bound of this interval and represents supporting evidence. Plausibility is the upper bound of the interval and represents non-refuting evidence. For instance, if a node N_k observes that one of its neighbors, say node N_j, has dropped packets with probability p, then node N_k has p degree of belief in the packet dropping behavior of node N_j and 0 degree of belief in its absence. The belief value with respect to an event α_i and observed by node N_k can be computed as the following.

$$bel_{N_k}(\alpha_i) = \sum_{e:\alpha_e \in \alpha_i} m_{N_k}(\alpha_e)$$

Here α_e are all the basic events that compose the event α_i, and $m_{N_k}(\alpha_e)$ stands for the view of the event α_e by node N_k. In this case, since node N_k merely get one single report of node N_j from itself, i.e., $\alpha_i \subset \alpha_i$. Therefore, we can derive that $bel_{N_k}(\alpha_i) = m_{N_k}(\alpha_i)$. Note that $\bar{\alpha_i}$ denotes the non-occurrence of the event α_i. Since the equation $pls(\alpha_i) = 1 - bel(\bar{\alpha_i})$ holds for belief and plausibility, we can further derive the following: $bel_{N_k}(N_j) = m_{N_k}(N_j) = p$ and $pls_{N_k}(N_j) = 1 - bel_{N_k}(\bar{N_j}) = 1$.

Parameter	*Value*
Simulation area	$150m \times 150m, 300m \times 300m$ $450m \times 450m, 600m \times 600m$
Number of nodes	50, 100, 150, 200
Transmission range	$60m, 90m, 120m$
Mobility pattern	*Random Waypoint*
Node motion speed	$5m, 10m, 20m$
Number of misbehaving nodes	5, 10, 20
Simulation time	900s

TABLE I
SIMULATION PARAMETERS

Given that belief indicates the lower bound of the uncertainty interval and represents supportive evidence, we define the combined packet dropping level of node N_j as the following.

$$pd_{N_j} = bel(N_j) = m(N_j) = \bigoplus_{k=1}^{K} m_{N_k}(N_j)$$

Here $m_{N_k}(N_j)$ denotes the view of node N_k on another node N_j. We can combine reports from different nodes by applying the Dempster's rule, which is defined as following.

$$m_1(N_j) \bigoplus m_2(N_j) = \frac{\sum_{q,r:\alpha_q \cap \alpha_r = N_j} m_1(\alpha_q) m_2(\alpha_r)}{1 - \sum_{q,r:\alpha_q \cap \alpha_r = \Phi} m_1(\alpha_q) m_2(\alpha_r)}$$

IV. PERFORMANCE STUDY

In this section, we examine the performance of the multi-dimensional trust management framework (*mTrust*), and its performance is compared to that of the baseline mechanism. The baseline mechanism that we choose here is the DST-based Outlier Detection mechanism (*DSTOD*), in which a lightweight trust management scheme was deployed [19].

A. Simulation Setup

We use GloMoSim 2.03 [26] as the simulation platform, and table I lists the parameters used in the simulation scenarios.

We use three parameters to evaluate the correctness and efficiency of our framework: *Correctness Rate* (CR), *Communication Overhead* (CO), and *Convergence Time* (CT). They are defined as follows.

$$CR = \frac{Number\ of\ True\ Misbehaving\ Nodes\ Detected}{Number\ of\ Nodes\ Picked\ As\ Suspects}$$

$$CO = \frac{Number\ of\ Packets\ for\ the\ Framework}{Total\ Number\ of\ Packets\ in\ the\ Network}$$

$$CT = Time\ taken\ to\ form\ a\ unique\ global\ view$$

Each simulation scenario has 30 runs with distinct random seeds, which ensures a unique initial node placement for each run.

B. Adversary Model

In MANETs, each node may choose to either cooperate with other nodes as well as follow all the network protocols, or their behaviors noticeably diverge from the behaviors of other nodes. Despite that the divergence can be caused by both malicious intents and out of ignorance, those abnormal

behaviors are both regarded as misbehaviors. A nodes that conducts some of all of those misbehaviors are regarded as an *adversary*. In the simulation scenarios, an adversary is able to conduct one or a set of misbehavior(s) chosen from the following misbehaviors: packet dropping, packet modification, malicious flooding of RTS frames, and intentional spreading of fake observations. Moreover, the adversaries are able to mix its misbehaviors at any rate if it choose to conduct multiple misbehaviors. Moreover, we assume that at most a small fraction of the nodes are adversaries, and all the nodes are placed in the simulation area in a random manner. Consequently, the fraction of the network area affected by them is bounded. Note that this assumption does not preclude that a few adversaries might surround a correct node at a certain point of time, even though collusion among adversaries is not considered in the simulation.

C. Simulation Results

The performance of *mTrust* is observed and compared to that of *DSTOD* in several distinct simulation scenarios. The simulation results show that: (1) In general, *mTrust* achieves a good performance in terms of correct detection of misbehaviors, quickly convergence to a unique global view of misbehaviors among all the nodes, and acceptable communication cost; (2) *mTrust* outperforms *DSTOD* especially in the scenarios in which there are some adversarial nodes, namely *Rumor Spreaders*, that perform nothing abnormal except deliberately propagating fake observations to other nodes; and (3) The utilization of the adaptive trust evolution model helps improve the overall performance of *mTrust* when compared to the previous trust evolution model in which all the trustworthiness evolve at a uniform (generally linear) pace. The simulation results are presented in details below.

1) Overall Performance of mTrust: To validate the proposed *mTrust* framework, we have observed the performance of *mTrust* as well as that of *DSTOD* in the following four scenarios: different number of nodes, different radio ranges, different percentage of misbehaving nodes, and different node motion speeds. Given that the simulation area remains unchanged in all these scenarios, we can observe from these scenarios the effect of node density, radio range, adversary percentage, and node mobility, respectively. Note that all the misbehaving nodes mix all the misbehaviors with a fixed rate in these experimental scenarios. In addition, there are five adversaries in the network except for the third scenario (i.e., different percentage of misbehaving nodes), in which the number of adversaries can be 5, 10 or 20. Therefore, there will not be any "pure" *rumor spreader* in this case. The simulation results are showed in the following Figure 5.

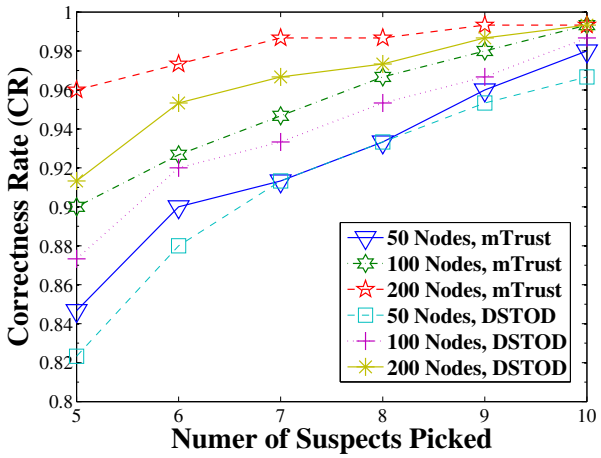
From Figure 5(a), we may find that when the node density is increased, *mTrust* yields a higher correctness rate. Moreover,it outperforms *DSTOD* when the node density is identical for both of them. Figure 5(b) illustrates that *mTrust* achieves a better performance when the radio range for each node is higher. This is true because the higher the radio range, the more likely it will be for each node to exchange observations with other nodes, which may lead to a more accurate view of misbehaviors. As is shown by Figure 5(c), the performance of *mTrust* and that of *DSTOD* will both be degraded when there are a larger amount of adversaries. However, *mTrust* is more resilient to a larger amount of adversaries than *DSTOD*. We can conclude from Figure 5(d) that the higher the node motion speed, the lower the performance will be. Nevertheless, *mTrust* is still able to derive an accurate view of misbehaviors even when the motion speed is rather high.

Besides the simulation results discussed above, we also observe the performance of *mTrust* and *DSTOD* in terms of communication overhead and convergence time. The simulation results show that *mTrust* yields a good performance in that it converges in a short period of time with a small communication overhead.

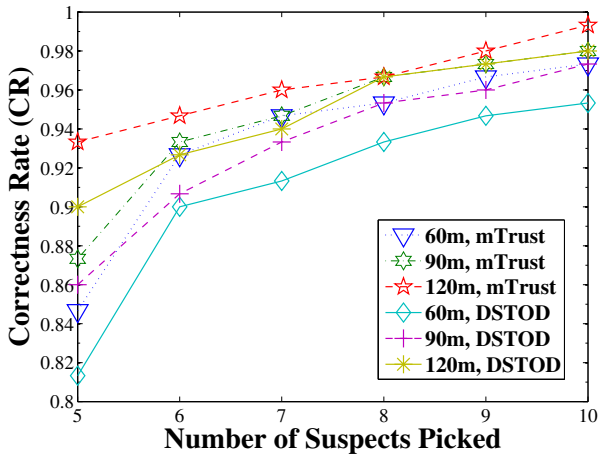
2) Effect of Rumor Spreading: As we have discussed in the previous section, intentional propagation of fake obsevations should be treated as one of the most dangerous misbehaviors in MANETs, because the adversaries can cause massive chaos by spreading rumors and consequently framing the good nodes. In this simulation scenario, there are some nodes that conduct nothing abnormal except delibrately exchanging fake observations with other nodes. In addition, there are some other nodes that conduct a mixture of various misbehaviors, including rumor spreading. We assume that the "pure" *rumor spreaders* randomly choose the victim that they want to frame, i.e., they will randomly generate fake observations to accuse their neighbors of various misbehaviors. The performance of *mTrust* and that of *DSTOD* are compared both in presence of "pure" *rumor spreaders* and without them. Figure 6 displays the simulation result for this scenario.

From Figure 6 we observe that both *mTrust* and *DSTOD* will be influenced in presence of the "pure" rumor spreaders. However, *mTrust* can yield a remarkably better performance than *DSTOD*. This finding is in coincidence with the nature of the two mechanisms. As we have discussed in the previous section, each dimension of trustworthiness in *mTrust* aims to keep track of the trustworthiness status of a node from one specific perspective. On the other hand, there is merely one single trustworthiness metric per each node in *DSTOD*. As a result, "pure" rumor spreaders may remain uncaught for a longer period of time in *DSTOD* due to the lack of a dedicated trustworthiness metric to keep track of the rumor spreading misbehavior. In addition, since all kinds of misbehaviors lead to one trustworthiness for each node, it is more likely that those asversarial nodes that conduct mixed misbehaviors will be detected first, which makes the "pure" rumor spreaders survive for a longer time. In contrast, there is one dimension of trustworthiness that is designated to track those "pure" rumor spreaders. Therefore, it is unlikely that the "pure" rumor spreaders can make use of other misbehaving nodes to hide themselves.

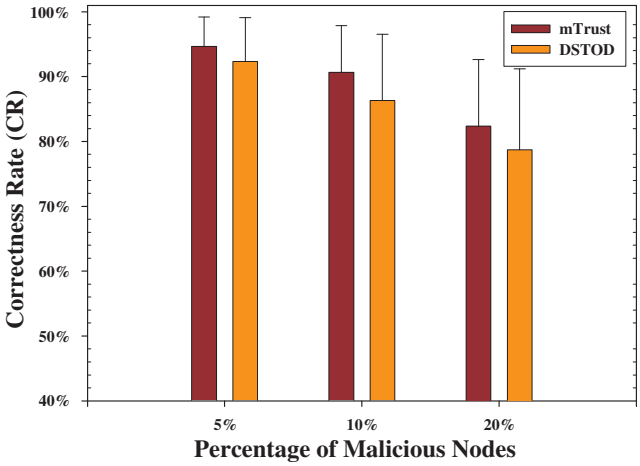
Similarly, if there are some adversaries that devote themselves to another misbehavior, such as packet dropping



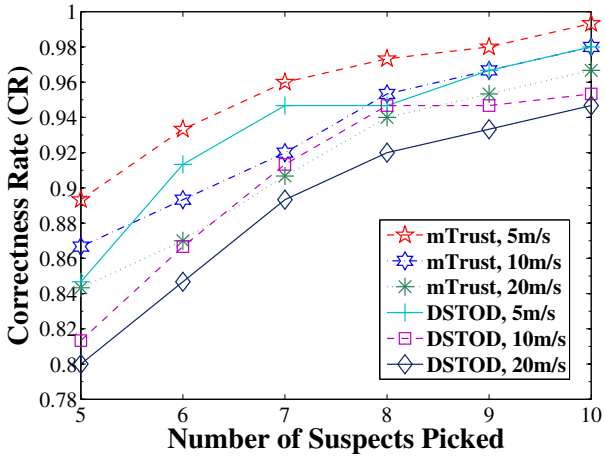
(a) Effect of Node Density



(b) Effect of Radio Range



(c) Effect of Adversary Percentage



(d) Effect of Node Mobility

Fig. 5. Performance of *mTrust* V.S. *DSTOD*

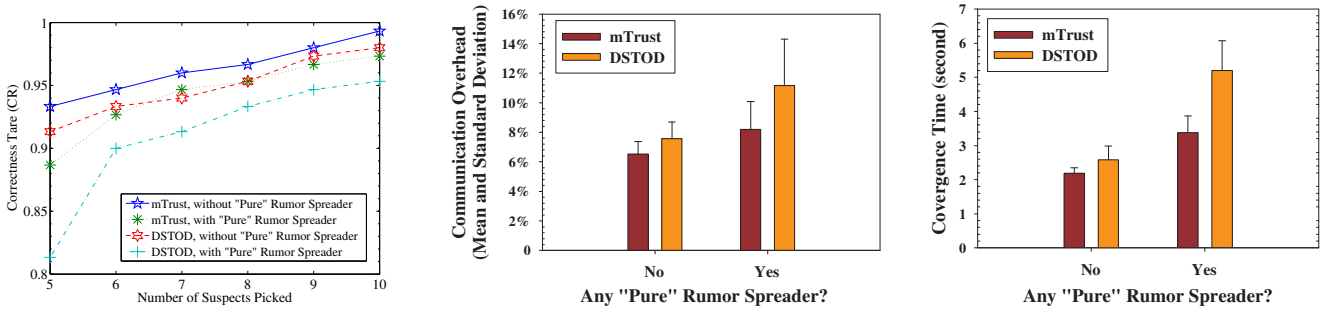
or packet modification, *mTrust* will outperform *DSTOD* because *mTrust* separately tracks the status of various misbehaviors in a more accurate manner.

3) Effect of Adaptive Trustworthiness Evolution Model: In this simulation scenario, we observe the performance of *mTrust* that utilizes either Adaptive Trustworthiness Evolution Model or the traditional trustworthiness evolution model, and presence of "pure" rumor spreaders is another factor that we may change here. Figure 7 demonstrates the simulation result for this scenario.

From Figure 7 we can find that the traditional trustworthiness evolution model suffers from the significant performance degradation caused by the emergence of "pure" rumor spreaders. On the other hand, the impact of those "pure" rumor spreaders to the adaptive model is quite limited. This is true because the adaptive model can properly decide the pace of punishment to various misbehaviors according to their nature. In this case, because rumor spreading is very harmful to the network, the exponential model is used to punish it. In this way, those "pure" rumor spreaders can be quickly excluded from the observation exchange process. As a result, *mTrust* that is equipped with the adaptive model can still yield a short convergence time even in presence of "pure" rumor spreaders. We should also note that the occurrence frequency as well as the actual content of the false observations will be carefully examined for each suspect before they are formally classified as "pure" rumor spreaders. In this way, we ensure that a node will not be mis-classified as a "pure" rumor spreader if it happens to obtain some incorrect observations from other nodes and then exchange them with others.

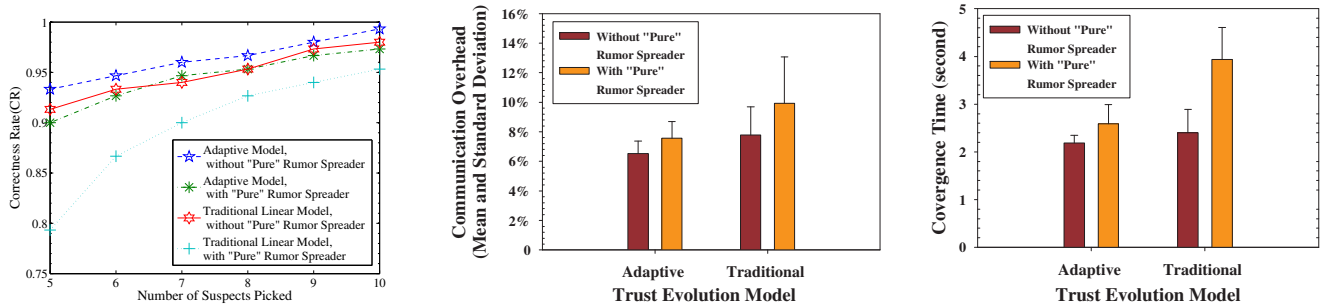
V. Conclusion

In this paper, a multi-dimensional trust management framework is proposed to better evaluate the trustworthiness of nodes in MANETs. Compared to the traditional trust management mechanisms, the trustworthiness of a node is judged from different perspectives, and each dimension of the trustworthi-



(a) (b) (c)

Fig. 6. Effect of Rumor Spreading on both *mTrust* and *DSTOD*



(a) (b) (c)

Fig. 7. Effect of Adaptive Trustworthiness Evolution Model on *mTrust*

ness is derived from various sets of misbehaviors according to the nature of those misbehaviors.

The key contributions of the proposed approach are: (1) A multi-dimensional trust management framework in which the notion of trustworthiness is further classified into several dimensions so that each dimension is able to precisely indicate whether or not a node is trustworthy in terms of one specific behavior that it should conduct, such as cooperation, well-behaving, and honest; and (2) an adaptive trust evolution model by which each dimension of the trustworthiness can be adjusted according to the features of the misbehavior to which the dimension is related, such as severity of the outcome, frequency of occurrence, and context in which the misbehavior occurs.

Simulation results obtained from multiple scenarios have proven that the proposed multi-dimensional framework is resilient to various misbehaviors including rumor spreading, and it can converge to an accurate view of misbehaviors for each node in MANETs with an acceptable communication cost.

REFERENCES

[1] P. Michiardi and R. Molva, "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security*. Deventer, The Netherlands, The Netherlands: Kluwer, B.V., 2002, pp. 107–121.

[2] L. Zhou and Z. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, no. 6, pp. 24–30, Nov/Dec 1999.

[3] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2000, pp. 275–283.

[4] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2000, pp. 255–265.

[5] S. Buchegger and J.-Y. Le Boudec, "Performance analysis of the confidant protocol," in *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM, 2002, pp. 226–236.

[6] G. Theodorakopoulos and J. S. Baras, "Trust evaluation in ad-hoc networks," in *WiSe '04: Proceedings of the 3rd ACM workshop on Wireless security*. New York, NY, USA: ACM, 2004, pp. 1–10.

[7] C. Zouridaki, B. L. Mark, M. Hejmo, and R. K. Thomas, "Robust cooperative trust establishment for manets," in *SASN '06: Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*. New York, NY, USA: ACM, 2006, pp. 23–34.

[8] Y.-A. Huang and W. Lee, "A cooperative intrusion detection system for ad hoc networks," in *SASN '03: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*. New York, NY, USA: ACM, 2003, pp. 135–147.

[9] A. Patwardhan, J. Parker, A. Joshi, M. Iorga, and T. Karygiannis, "Secure routing and intrusion detection in ad hoc networks," in *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications, 2005. PerCom 2005.* IEEE, March 2005, pp. 191–199.

[10] J. Parker, A. Patwardhan, and A. Joshi, "Cross-layer analysis for detecting wireless misbehavior," in *Proceedings of the Third IEEE Consumer Communications and Networking Conference, 2006. CCNC 2006.*, vol. 1. IEEE, Jan. 2006, pp. 6–9.

[11] S. Buchegger and J.-Y. Le Boudee, "Self-policing mobile ad hoc networks by reputation systems," *IEEE Communications Magazine*, vol. 43, no. 7, pp. 101–107, July 2005.

[12] P.-W. Yau and C. J. Mitchell, "Security vulnerabilities in ad hoc networks," in *Proceedings of the 7th International Symposium on Communication Theory and Applications*, 2003, pp. 99–104.

[13] H. Deng, Q.-A. Zeng, and D. Agrawal, "Svm-based intrusion detection system for wireless ad hoc networks," in *Proceedings of 2003 IEEE 58th Vehicular Technology Conference, 2003. VTC 2003-Fall.*, vol. 3, Oct. 2003, pp. 2147–2151.

[14] C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt, "A specification-based intrusion detection system for aodv," in *SASN '03: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*. New York, NY, USA: ACM, 2003, pp. 125–134.

[15] L. Anderegg and S. Eidenbenz, "Ad hoc-vcg: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents," in *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2003, pp. 245–259.

[16] Y. Xue and K. Nahrstedt, "Providing fault-tolerant ad hoc routing service in adversarial environments," *Wirel. Pers. Commun.*, vol. 29, no. 3-4, pp. 367–388, 2004.

[17] M. Kefayati, H. R. Rabiee, S. G. Miremadi, and A. Khonsari, "Misbehavior resilient multi-path data transmission in mobile ad-hoc networks," in *SASN '06: Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*. New York, NY, USA: ACM, 2006, pp. 91–100.

[18] W. Li, J. Parker, and A. Joshi, "Security through collaboration in manets," in *Proceedings of 4th International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom 2008*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST), vol. 10. Springer, 2008, pp. 696–714.

[19] W. Li and A. Joshi, "Outlier detection in ad hoc networks using dempster-shafer theory," in *Proceedings of the Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09.*, May 2009, pp. 112–121.

[20] W. Li, A. Joshi, and T. Finin, "Policy-based malicious peer detection in ad hoc networks," in *Proceedings of the International Conference on Computational Science and Engineering, 2009. CSE '09.*, vol. 3, Aug. 2009, pp. 76–82.

[21] S. Buchegger and J.-Y. L. Boudec, "A robust reputation system for mobile ad-hoc networks," in *Proceedings of P2PEcon*, 2003.

[22] Q. He, D. Wu, and P. Khosla, "Sori: a secure and objective reputation-based incentive scheme for ad-hoc networks," in *Proceedings of 2004 IEEE Wireless Communications and Networking Conference, WCNC '04.*, vol. 2, March 2004, pp. 825–830 Vol.2.

[23] S. Buchegger and J.-Y. L. Boudec, "The effect of rumor spreading in reputation systems for mobile ad-hoc networks," in *Proceedings of WiOpt 2003: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.

[24] A. Patwardhan, A. Joshi, T. Finin, and Y. Yesha, "A data intensive reputation management scheme for vehicular ad hoc networks," in *Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems - Workshops, Mobiquitous '06.*, July 2006, pp. 1–8.

[25] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ, USA: Princeton University Press, 1976.

[26] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: a library for parallel simulation of large-scale wireless networks," *ACM SIGSIM Simulation Digest*, vol. 28, no. 1, pp. 154–161, 1998.