Distributed and Parallel Databases manuscript No.
(will be inserted by the editor)

CAST: Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies

Wenjia Li · Anupam Joshi · Tim Finin

Received: 2/6/2012 / Accepted: date

Abstract Due to lack of pre-deployed infrastructure, nodes in Mobile Ad-hoc Networks (MANETs) are required to relay data packets for other nodes to enable *multi-hop* communication between nodes that are not in the radio range with each other. However, whether for selfish or malicious purposes, a node may refuse to cooperate during the network operations or even attempt to interrupt them, both of which have been recognized as misbehaviors. Significant research efforts have been made to address the problem of detecting misbehaviors. However, little research work has been done to distinguish truly malicious behaviors from the faulty behaviors. Both the malicious behaviors and the faulty behaviors are generally equally treated as misbehaviors without any further investigation by most of the traditional misbehavior detection mechanisms. In this paper, we propose and study a Context-Aware Security and Trust framework (CAST) for MANETs, in which various contextual information, such as communication channel status, battery status, and weather condition, are collected and then used to determine whether the misbehavior is likely a result of malicious activity or not. Simulation results illustrate that the CAST framework is able to accurately distinguish malicious nodes from faulty nodes with a limited overhead.

Keywords Mobile Ad-hoc Network · Security · Trust · Misbehavior detection · Context awareness · Policy

Wenjia Li
Department of Computer Sciences
Georgia Southern University
Statesboro, GA 30460
Tel.: +1-912-478-7392
E-mail: wenjiali@georgiasouthern.edu

Anupam Joshi · Tim Finin
Department of CSEE
University of Maryland, Baltimore County
Baltimore, MD 21250
E-mail: {joshi,finin}@cs.umbc.edu

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

1 Introduction and Motivation

A Mobile Ad-hoc Network (MANET) is a self-configuring network of mobile devices that are connected by wireless links. In a MANET, each device is willing to serve as a router and share its transmission power with other devices because it is required to forward traffic that is irrelevant to its own interest. Mobile Ad-hoc NETworks (MANETs) have a variety of both civilian and military applications, ranging from emergency disaster relief personnel coordinating rescue efforts after a hurricane, earthquake or brush fire to soldiers exchanging information for situational awareness on the battlefield. Other possible applications include mobile healthcare system, real-time traffic alert propagation via vehicular networks, and Cyber-Physical System (CPS).

Security is a key concern in MANETs because the nodes in MANETs are generally more susceptible to various threats than those in the traditional wired networks. From a security perspective, security systems in MANETs differ significantly from those in the traditional wired networks because of the following features of MANETs.

- Open and error-prone transmission medium: data in MANETs are transmitted via Radio Frequency (RF) broadcasts, and The open nature of RF signal makes the transmitted data extremely susceptible to both transmission errors and intentional tampering.
- Absence of pre-deployed infrastructure: as a centralized network infrastructure is no longer feasible to MANETs, cooperation among the mobile nodes becomes a most critical precondition for the security systems in MANETs, and these security systems in MANETs are required to be more resilient to uncooperative behaviors than those in wired networks.
- Rigorous power constraint: due to the limited battery power, the transmission range among the nodes is generally restricted, which makes it very difficult for each node to obtain a global view on what are happening in the whole network only from its own observations. Hence, each node also needs to rely on the observations from other nodes to fully understand what are happening in the network (for example, which nodes are behaving normally and which nodes are not)[1–6].
- Highly dynamic network topology: being equipped with wireless transmission devices, the mobile nodes in MANETs are free to make any arbitrary movement, which can occasionally interrupt communication because nodes may move out of the transmission range of each other from time to time during communication processes.

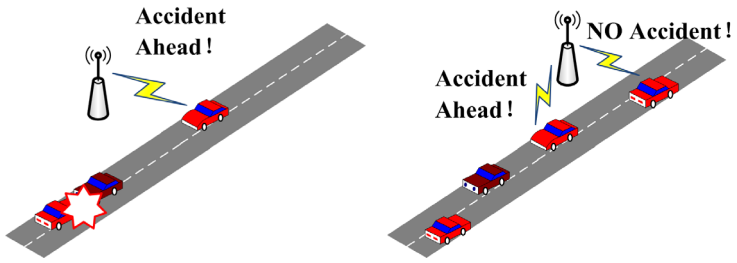
Because of the features listed above, MANETs are extremely vulnerable to a variety of node misbehaviors when compared to the traditional wired networks. Therefore, security is one of the most important challenges for MANETs, and the traditional security solutions for the wired networks may not be directly applied to MANETs.

Since node misbehaviors can do great harm to MANETs, numerous security solutions have been proposed to detect and mitigate those misbehaviors from a

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

variety of perspectives. Among these security solutions, misbehavior detection method is a well-known countermeasure to fight against node misbehaviors [7,8,2]. The majority of existing misbehavior detection methods merely aim to identify the misbehaving nodes without further investigating the cause of those misbehaviors. However, many of these misbehaviors may also occur due to environmental and mobility related reasons, not just malicious intent. It is straight forward that malicious behaviors are far more dangerous than the faulty behaviors, because the goal of the malicious attackers is to disturb the network operations by carrying out the misbehaviors, whereas faulty nodes do not aim to intentionally disrupt the network and their effects are generally self limiting. Hence, it is essential to properly distinguish malicious attackers from faulty nodes.

Let us take the traffic monitoring system as an example, which is depicted in Figure 1. Present generation monitoring systems are based on ground sensors and cameras. However, with increasing computing and communication capabilities embedded in vehicles, their onboard sensors themselves can be used to monitor traffic. From Figure 1(a), we find that a vehicle observes an accident ahead, and it reports this accident to the system. Therefore, the traffic alarm shown in Figure 1(a) is true. In contrast, Figure 1(b) shows two conflicting traffic alarms. Given that there is no accident in this scenario, the vehicle that reports accident to the system is misbehaving. However, we need to further investigate the context to decide if this vehicle is *faulty* or *malicious*. For instance, if the vehicle is traveling too fast or there is a blizzard, then the sensor on the misbehaving vehicle may malfunction and send the incorrect accident alert without any malicious intent.



(a) True Traffic Alarm (b) Conflicting Traffic Alarm

Fig. 1 True Alarm VS False Alarm in Traffic Monitoring System

Trust management scheme is also a widely-used security solution that is closely coupled with the misbehavior detection scheme to cope with node misbehaviors. Because it is quite useful to evaluate a node's behaviors and determine if it is trustworthy in terms of how cooperative it is, trust management schemes have become a powerful tool to deal with node misbehaviors. A variety of trust management mechanisms have been extensively studied during the past decade, such as the mechanisms discussed in [1], [9], and [10]. Most

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

of these trust management mechanisms model the trust of a node in one dimension, i.e., all available evidences and observations are used to calculate a single scalar trust metric for each node. However, a single scalar trust may not be expressive enough to accurately describe whether a node is trustworthy or not in many complicated scenarios. Fig. 2 shows an example scenario in which a single scalar trust is not expressive enough.

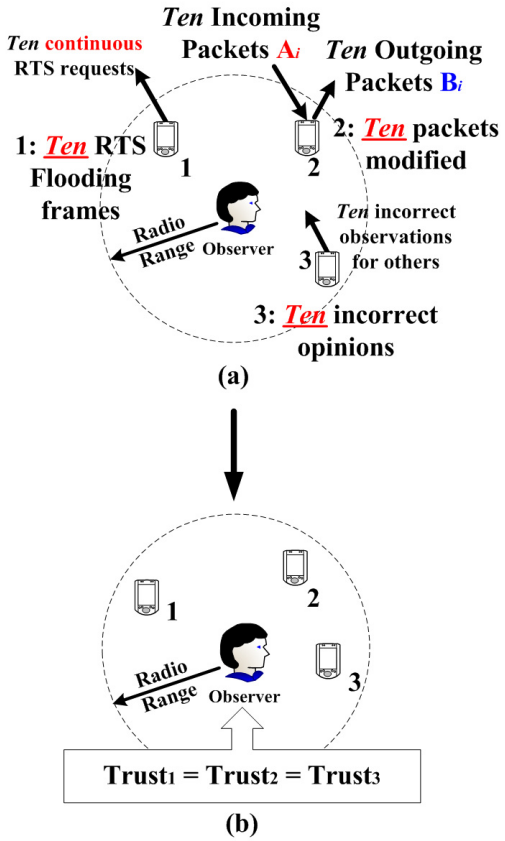


Fig. 2 An example scenario where a single scalar trust is NOT expressive enough

From Fig. 2, we see that in the first step (a), the observer collects and records the misbehaviors that are conducted by node 1, 2, and 3. The observation results illustrate that node 1, 2, and 3 have modified packets, spread incorrect opinions regarding others (for example, intentionally accuse other nodes of dropping packets even if they have not done so) and sent continuous Request-To-Send (RTS) frames at a same amount of 10, respectively. Suppose that these three types of misbehaviors are punished at the same rate when the trustworthiness of each node is evaluated. Then, in the next step (b), the

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

observer may draw a conclusion that all these three nodes are equally trustworthy. As a result, the observer will treat node 2 and node 3 equally when it needs to determine which node to forward packets as well as which node it should believe when exchanging opinions. However, it is obvious that the trustworthiness of node 2 and node 3 is not equivalent when it comes to both packet forwarding and opinion exchanging. Hence, we can safely infer that it is neither accurate nor effective to merely use one single scalar when the trustworthiness of a node is evaluated.

In this paper, a Context-Aware Security and Trust framework (a.k.a *CAST*) is proposed and evaluated to help better secure MANETs. In CAST, the mobile nodes in MANETs observe and record the abnormal behaviors of their neighbors in a manner similar to existing methods [8,7,11]. In contrast to most existing approaches however, each peer also simultaneously collects the context information within which the abnormal behaviors occur. When each peer decides if a node is malicious based on observing abnormal behaviors, it factors in the context information in a manner specified by a policy. In other words, the policy specifies, based on the context, how "abnormal" is defined. Moreover, a *multi*-dimensional trust management scheme is deployed in SMART to better assess the trustworthiness of nodes in MANETs. Compared to the traditional *single*-dimensional trust management mechanisms [1,9,10], the trustworthiness of a node is judged from different *perspectives* (i.e., *dimensions*), and each dimension of the trustworthiness is derived from different sets of misbehaviors according to the nature of those misbehaviors.

The rest of the paper is organized as follows: In Section 2, we briefly review the literature on misbehavior detection, trust management, as well as policies for MANETs. Next, we present the system model as well as the adversary model in Section 3, followed by a detailed discussion to the proposed CAST framework in Section 4. Then, a comprehensive simulation study is given in Section 5 to evaluate the performance of the SMART framework, and the paper is finally concluded in Section 6.

2 Related Work

In recent years, there has been a rich literature on 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 discussed separately in this section.

2.1 Misbehavior Detection for Ad hoc Networks

The term *misbehavior* generally refers to a group of abnormal behaviors that deviates from the set of behaviors that each node is supposed to conduct in MANETs [12]. In general, misbehaviors can occur at each layer in MANETs, such as (1) malicious flooding of the RTS frames in the MAC layer, (2) drop,

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

modification, and misroute to the packets in the network layer, and (3) deliberate propagation of fake opinions regarding the behaviors of other nodes in the application layer.

Moreover, node misbehaviors may range from lack of cooperation to active attacks aiming to cause Denial-of-Service (DoS) and interruption of the network operations. According to [13], there are four types of misbehaviors in ad hoc networks, namely *failed node behaviors*, *badly failed node behaviors*, *selfish attacks*, and *malicious attacks*, respectively. 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. For instance, because of the limited battery power that each node possesses, a *selfish* node may choose not to cooperate with other nodes so as to preserve its own battery power [14]. In other words, when a *selfish* node is requested to forward some data packets for other nodes, it may choose to drop either a part or all of the incoming packets. By this means, it can save its battery power and thus transmit some extra packets for the sake of itself. On the contrary, the *malicious* nodes aim to intentionally disturb the network services, and they may deliberately drop, modify or misroute packets while their primary concern is not battery lives [15]. Regardless of the intents by which the node misbehaviors are induced, they are both harmful to a currently healthy MANET.

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 mobile node due to the lack of a fixed infrastructure, such as [7,16,17]. 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. Then, each node may cooperate with other nodes to further refine the detection results from time to time. On the other hand, Huang et al. [18] proposed a cooperative intrusion detection framework, in which clusters are formed and a node in each cluster will act as the cluster head in turn and coordinate amongst all the cluster members for the intrusion detection process. In addition, Parker et al. [11] proposed a cross-layer intrusion detection method in which evidences for misbehaviors are collected and then combined at multiple layers. By this means, the observations from multiple layers are integrated in case that they are related to the same misbehavior, and the misbehaviors can be *amplified* in presence of various ambient noises, such as abnormal behaviors caused by radio interference and congestion.

Some research efforts have been made to apply machine learning techniques to detect node misbehaviors [7,16,19,20]. In these approaches, misbehaviors are regarded as anomalies, and we need to first collect and label training data to train a classifier. On the contrary, we use the policy rules to define anomalies,

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

and the use of outlier detection makes it easier and quicker to be deployed in practice than the approaches based on machine learning techniques.

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 these internal resources to disturb the routing services so as to make part of or the entire network unreachable. Marti et al. [8] introduced two related techniques, namely *watchdog* and *pathrater*, to detect and isolate misbehaving nodes, which are nodes that do not forward packets for others. There are also some other solutions that aim to cope with various routing misbehaviors [21–23].

2.2 Trust Management for MANETs

The main goal of trust management is to evaluate behaviors of other nodes and consequently build a reputation for each node based on the behavior assessment. The reputation can then be used to decide the trustworthiness for other nodes, make choices on which nodes to cooperate with, and even take action to punish an untrustworthy node if needed.

In general, a trust management scheme relies on two types of observations to evaluate the node behaviors. The first kind of observation is named as *first-hand* observation, or in other words, *direct* observation [24]. First-hand observation is the observation that is directly made by a node itself. 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 generally related to overhead, false report and collusion [25, 26].

In [1], 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. [14] 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,

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

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. [27] 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.

Ren et al. [28] proposed a node evaluation scheme in which each node evaluates the trustworthiness of its neighbors with the assistance of trustworthy neighboring nodes. More specifically, the second-hand observations may be obtained from only a subset of the node's neighbors, and these selected neighbors are regarded as trustworthy sources with respect to the opinions toward other nodes.

In our previous work [3–6], we address the need for node behavior assessment by deploying a simple yet effective trust management scheme. In this scheme, the trustworthiness of each node is represented by one single scalar value, and all the observation results are used to derive this single scalar trustworthiness.

2.3 Policies for Security in Distributed Systems

According to Sloman, policies define a relationship between subjects and targets [29]. Policy-based security is often used in systems where flexibility is required as users, services and access rights change frequently, such as mobile ad-hoc networks and other large-scale distributed systems. In these distributed systems, it is essential to ensure that all the heterogeneous entities behave appropriately. Therefore, policy based security should be the most effective mechanism for distributed systems, because it is possible to specify how different entities act without modifying their internal mechanisms [30].

Multiple policy languages have been studied in the past decade, such as Extensible Access Control Markup Language (XACML) [31] and the *Rei* policy language [30]. XACML [31] is a language in XML for expressing access policies. XACML allows control over actions and supports resolution of conflicts. On the other hand, Rei is a policy language designed for pervasive computing

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

applications that is based on deontic concepts and grounded in a semantic language.

3 System and Adversary Model

3.1 System Model

In this paper, we view a MANET as a set Δ of N mobile nodes, that is, $|\Delta| = N$. The network size N may be dynamically changing while nodes join, leave, or experience a variety of failures (such as communication interruption and exhausted battery power). Every node $A \in \Delta$ has a unique ID, which may generally be represented by its network-layer address.

The term *node* is defined as a system entity in MANETs that is capable of observing the behaviors of other nodes within its radio transmission range, exchanging these observations with other nodes within its radio transmission range, and also sensing the context such as weather condition, battery status, and the motion speed. We define 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.

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

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 types of 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.

3.2 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*.

Our goal in this paper is to contrive a sound security and trust framework to secure MANETs. Therefore, we assume that the adversary aims to disrupt network operations by conducting a variety of misbehaviors, 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. Even if it is also important to mitigate the attacks that generally target to compromise key management protocols in MANETs (such as the man-in-the-middle attack discussed in [32]), these key management attacks are beyond the scope of this paper.

We further assume that the adversary is able to mix its misbehaviors at any ratio if it choose to conduct multiple misbehaviors at the same time period. In addition, the adversary may alter the ratio of each misbehavior from time to time, and it can carry out the set of misbehaviors for any arbitrary length of time. For instance, an adversary A may determine at time t_1 that it should equally conduct the four types of misbehaviors (i.e., RTS flooding, packet dropping, packet modification, and packet misroute); while at time t_2, A changes its attack model to solely perform RTS flooding attack.

Moreover, we assume that at most a small fraction of the nodes are adversaries, and all the nodes are placed in a random manner. Consequently, the fraction of the network area affected by adversaries 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 here.

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

4 The Context-Aware Security and Trust Framework

In this section, we present the CAST framework in details. The goal of the CAST framework is to properly assess the behaviors of each mobile nodes in different contexts using policies.

4.1 Framework Overview

In the policy and trust driven framework, there are four major functional units, namely Data Collection, Policy Management, Misbehavior Detection, and Trust Management. Figure 3 illustrates the CAST framework.

The Data Collection unit is mainly responsible for collecting contextual data and node behavioral data, and then sending either of them to the Policy Management unit, the Malicious Node Detection unit, or the Trust Management unit.

The trustworthiness of each node is assessed by the Trust Management unit, in which both direct observations (made by a node itself) and indirect observations (obtained from another node) are both taken into account to evaluate how trustworthy a node is. Note that here the trustworthiness of mobile nodes are assessed from three distinct perspectives, by which the trustworthiness can be appraised in a more accurate manner.

The Policy Management unit is used to collect and record various contextual information, and then enforce the corresponding security policies so that both the Malicious Node Detection unit and Trust Management unit can make use of the contextual information when they identify the malicious nodes.

We use the Malicious Node Detection unit to identify misbehaviors and then distinguish truly malicious nodes from malfunctioning nodes using the security policies.

4.2 Data Collection

In the CAST framework, two types of data are sensed and collected: node behaviors and contextual information. The node behaviors are used by both the Malicious Node Detection and the Trust Management units to identify misbehaving nodes and evaluate nodes' trustworthiness. The contextual information is used by the Policy Management unit to specify and enforce policies that can then be used to capture the truly malicious nodes among those misbehaving nodes.

With the gradually wider deployment of various sensors in our daily lives, it is easier to better understand the context that surrounds us. For instance, various smart phone platforms, such as the Android phones, provide a wide variety of embedded sensors that can be used to collect and understand the context [33]. In addition, the deployment of vehicle onboard sensors makes it even more convenient to collect the contextual information from additional

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

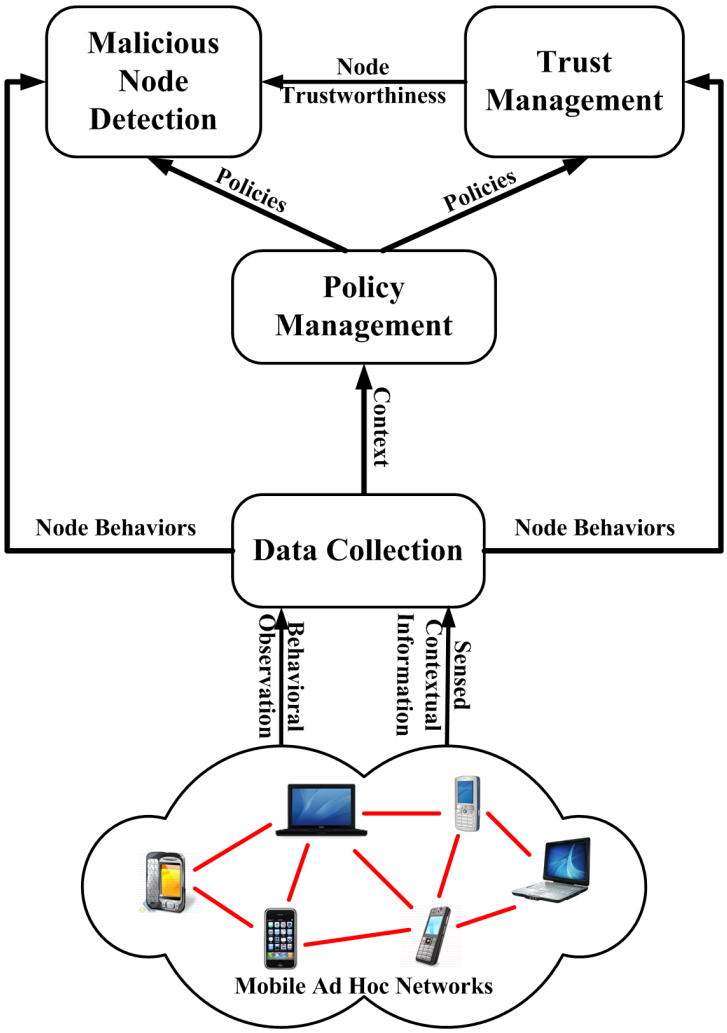


Fig. 3 Schematic Diagram of the CAST Framework

sources [34]. Therefore, it is critical for us to find a feasible way to better represent and understand the rich context brought by various sensors. We believe that using policies is such a feasible way.

4.3 Policy Management

In the Policy Management unit, all the contextual information will be used in policies. For example, as is shown in Figure 1(b), if a vehicle is found to report inconsistent traffic information, then the contextual information is used in this

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

Table 1 Dynamic environmental data received from sensors. It consists of temperature, weather conditions, location details - latitude, longitude, altitude and speed.

```
MANET:Sensor_Device a MANET:Sensor ;
        MANET:has_sensor_id "1" ;
        MANET:has_sensor_type "X" ;
        MANET:has_sensed_information MANET:Sensed_Data.
MANET:Sensed_Data a MANET:Sensor_Information ;
        MANET:has_temperature "20F" ;
        MANET:has_speed "5M/S" ;
        MANET:has_location_information MANET:Location_Data ;
        MANET:has_weather_information MANET:Weather_Data .
MANET:Location_Data a MANET:Location_Information ;
        MANET:has_latitude "39.253571";
        MANET:has_longitude "-76.714191";
        MANET:has_altitude "249".
MANET:Weather_Data a MANET:Weather_Information ;
        MANET:has_weather_condition "4" ;
```

case to determine whether these inconsistent traffic alerts are possibly caused by environmental factors or not.

In the CAST framework, we define a set of comprehensive policy rules. Some of them come from human experts, and others arise naturally from well known facts in wireless networks. For example, it is well understood that data transmission in wlreless network is very likely to be interrupted if the ambient noise is high. Thus, we define a policy rule saying that "If ambient noise is high, then lower the punishment for packet dropping." Also, we will update the rule set periodically if we find that there is an important rule that will significantly influence the performance of the CAST framework.

In the experiments, we use Jena [35] to describe the policy rules, because it is one of the commonly used tools for reasoning over OWL/RDf data. Table 1 describes an example of various contextual information collected by sensors and sent to the Data Collection unit, such as the current weather conditions, geolocation, temperature, and motion speed. The contextual information is then reported to the Policy Management unit. Then Policy Management unit analyses the reported contextual information and uses policies to determine whether the vehicle is intentionally reporting incorrect traffic alerts or the current environmental conditions cause the malfunctioning vehicle report those incorrect traffic alerts.

The system can have multiple policies to consider the effects of various environmental factors. For instance, policies can be declared as (i) *If surrounding temperature is beyond range 0F-120F then there is a possibility of faulty behavior*, (ii) *If the motion speed is more than 20 M/S then there is a possibility of faulty behavior*, (iii) *If the current weather conditions are either of heavy raining, snowing or foggy then there is a possibility of faulty behavior* and (iv) *If the altitude is higher than 2000 feet, weather conditions are snowing and temperature is below 32F then there is a possibility of faulty behavior.* These

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

Table 2 Policy to report the possibility of faulty behaviors if surrounding temperature is beyond range 0F-120F.

```
[TemperatureRule:
        (?sensorDevice a MANET:Sensor)
        (?sensorDevice MANET:has_sensed_information ?sensedData)
        (?sensedData MANET:has_temperature ?temperature)
        lessThan(?temperature, 0) greaterThan(?temperature, 120)
        ->
        (?sensorDevice MANET:faulty_device “true”)
]
```

Table 3 Policy to report the possibility of faulty behaviors if the motion speed is too high.

```
[SignalStrengthRule:
        (?sensorDevice a MANET:Sensor)
        (?sensorDevice MANET:has_sensed_information ?sensedData)
        (?sensedData MANET:has_speed ?speed)
        greaterThan(?speed, 20)
        ->
        (?sensorDevice MANET:faulty_device “true”)
]
```

Table 4 Policy to report the possibility of faulty behaviors if current weather conditions are either of Heavy raining, Snow or Foggy.

```
#For convenience, conditions are mapped to numerical values as
#Clear = 1, Sunny = 2, Heavy raining = 3, Heavy snow = 4, Foggy = 5
[WeatherConditionsRule:
        (?sensorDevice a MANET:Sensor)
        (?sensorDevice MANET:has_sensed_information ?sensedData)
        (?sensedData MANET:has_weather_information ?weatherData)
        (?weatherData a MANET:Weather_Information)
        (?weatherData MANET:has_weather_condition ?weatherCondition)
        greaterThan(?weatherData, 2)
->
        (?sensorDevice MANET:faulty_device “true”)
]
```

policies are represented in Jena's rules syntax specification in Table 2, Table 3, Table 4 and Table 5. Note that these are just some examples of the policy rules that we can declare, and the policy rules can be written over any set of those environmental factors.

Based on the example policy rules shown in these tables, the Policy Management unit can determine whether the misbehaviors are caused by environmental factors, or they are deliberately conducted by adversaries. Then, this

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

Table 5 Policy to report the possibility of faulty behaviors in case of a high altitude.

```
#Mobile node can exhibit faulty behaviors if altitude is greater than
#2000 ft,weather conditions are snowing and temperature is below 32F.
[AltitudeRule:
        (?sensorDevice a MANET:Sensor)
        (?sensorDevice MANET:has_sensed_information ?sensedData)
        (?sensedData MANET:has_weather_information ?weatherData)
        (?sensedData MANET:has_location_information ?locationData)
        (?weatherData MANET:has_weather_condition ?weatherCondition)
        (?sensedData MANET:has_altitude ?altitude)
        (?sensedData MANET:has_temperature ?temperature)
        equal(?weatherData, 4) lessThan(?temperature, 32)
        greaterThan(?altitude, 2000)
->
        (?sensorDevice MANET:faulty_device "true")
]
```

conclusion is used by both the Trust Management unit and the Malicious Node Detection unit to identify the truly malicious nodes.

4.4 Malicious Node Detection

The goal of the Malicious Node Detection unit is to properly identify the malicious nodes in MANETs by using the distributed misbehavior detection mechanism as well as the policies that have integrated the contextual information.

In this unit, we use the gossip-based outlier detection algorithm to identify the misbehaving nodes. Outliers are generally defined as data points that are very different from the rest of the data with respect to some measure [36]. The basic observation is that misbehaving nodes generally behave abnormally from those normal nodes. Thus, we can detect those misbehaving nodes by means of outlier detection.

The gossip-based outlier detection algorithm contains the following four steps, namely local view formation, local view exchange, local view update, and global view formation.

The first step of this algorithm is the formation of local views. The mobile nodes monitor and record the possible abnormal behaviors of other nodes within their radio range. Each node generates its local view of outliers based on their own observations. Once all the nodes form their local views, they will broadcast the local views to all of their immediate neighbors, i.e., all the nodes that are one hop away from them. Upon reception of a local view from another node, the recipient will update its local view based on the received view.

Here we use the Dempster-Shafer Theory of evidence (DST) [37] to combine the local view and the received external view. The DST-based view combination method is shown in Algorithm 1. Note that n_i stands for the i-th node

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

in the MANET. V_i denotes the initial view of n_i, and V_i' denotes the updated view of n_i.

Algorithm 1 Update of Local View for node i Using the Dempster-Shafer Theory (DST)

Input of n_i: V_i
Output of n_i: V_i'
Upon reception of V_k from node n_k:
if $V_i \neq V_k$ **then**

1. merge V_i and V_k according to the following rules:
   - if node m is in **BOTH** V_i **AND** V_k, then calculate the updated value U_i of the corresponding columns for node m in BOTH V_i and V_k using the Dempster's rule of combination, and store U_i to an intermediate list $TEMP_i$ as an entry.
   - if node m is in **EITHER** V_i **OR** V_j , but **NOT BOTH**, then add a virtual entry of node m to the view that previously does not contain m, and set all the columns of this virtual entry as 0. Then calculate the updated value U_i of the corresponding columns for node m in BOTH V_i and V_k using the Dempster's rule of combination, and store U_i to an intermediate list $TEMP_i$ as an entry.
2. calculate the top k outliers from $TEMP_i$, and assign these k top outliers to V_i'.
3. broadcast V_i' to all of its immediate neighbors (i.e., number of hop = 1).

else keep V_i unchanged, and do not send any message out.
end if

Note that unlike the traditional gossiping algorithm, the more the nodes that accept the same view of outliers, the less the number of new messages that are sent out. Ultimately, when all the nodes hold the same view of outliers, the algorithm halts, and the view that all the nodes hold is regarded as the global view of outliers.

The pseudo-code of the gossip-based outlier algorithm is given in Algorithm 2 and uses the same notation as described earlier. In addition, GV denotes the ultimate global view.

Algorithm 2 Gossip-based Outlier Detection Algorithm for MANETs

Input of n_i: V_i
Output of n_i: GV
For each node n_i
broadcast V_i to all of its immediate neighbors
Upon reception of V_k from its immediate neighbor n_k:
Invoke Algorithm 1
When no more message exchange occurs:
$\forall i, V_i = GV$

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

4.5 View Combination

As we have discussed in [4], Dempster-Shafer theory of evidence (DST) [37] 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 $pls_{N_k}(N_j) = 1 - bel_{N_k}(\bar{N_j}) = 1$.

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.

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

4.6 Trust Management

In the Trust Management unit, the trustworthiness of a node N_k is 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* (COT), *Behavioral Trust* (BET), and *Reference Trust* (RET), respectively. The three dimensions of trustworthiness are demonstrated in Fig. 4.

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

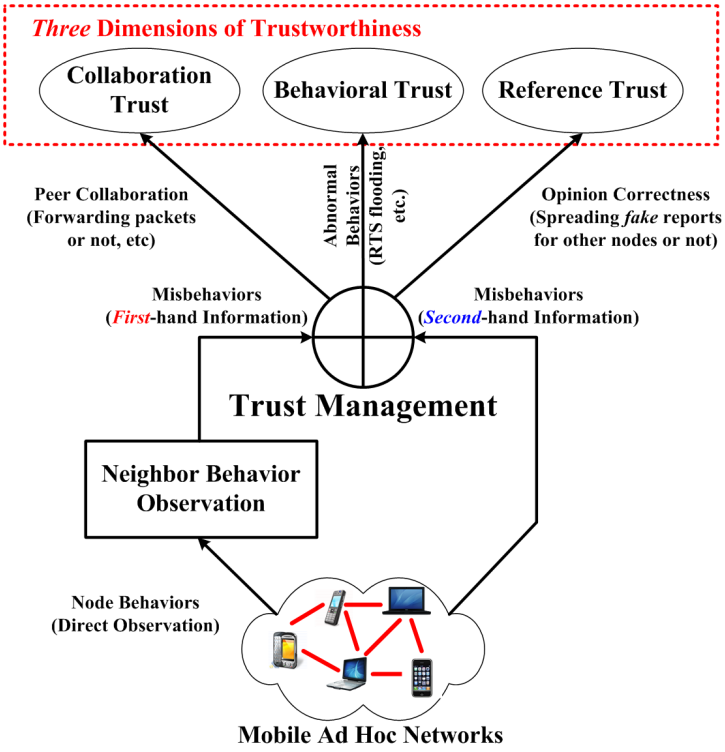


Fig. 4 The Three Dimensions of Trustworthiness

From Fig. 4 we find that COT ($\theta_k^{(1)}$) 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. BET ($\theta_k^{(2)}$) is derived by the amount of abnormal behavior that N_k has conducted, including packet modification, packet misroute or RTS flooding attack. RET ($\theta_k^{(3)}$) is generally computed based on the correctness of the observation results that N_k spreads. For instance, if N_k has been witnessed repeatedly sending fake observations to its neighbors, then $\theta_k^{(3)}$ should be assigned a very 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 those nodes themselves.

Note that misbehaviors can be caused by both malicious intent or environmental factors. 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 nature of misbehaviors into account, we have developed an *adaptive* trustworthiness

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

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 distinguish trustworthy neighbors from untrustworthy ones. As a result, *Reference Trust* ($\theta_k^{(3)}$) should decrease at the highest rate when compared to both COT and BET. 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.

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

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

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

Not only can the trust evolution model differs for different dimensions of trustworthiness, it can also alter for the same dimension in different contexts. 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 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.

5 Performance Evaluation

In this section, we examine the performance of the CAST Framework. We obtain some simulation results to evaluate the performance of the framework. In addition, we declare and execute some example policies on mobile platforms such as Android phones. They are representative of the type of computing capabilities likely to be present in the possible actual deployment of MANETs. In this way, we know that the CAST framework works well on both the simulation platform and real devices.

5.1 Simulation Results and Analysis

We use GloMoSim 2.03 [38] as the simulation platform, and table 6 lists the parameters used in the simulation scenarios. In the simulation, each node

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

Table 6 Simulation Parameters

Parameter	*Value*
Simulation area	$600m \times 600m$
Num. of nodes	50, 100, 200
Transmission range	$120m$
Node placement	*Random*
Num. of bad nodes	5, 10, 20
Simulation time	900s

collects various contextual information, such as motion speed, temperature, altitude, etc., so that it can better understand the context and thus evaluate the trustworthiness of its neighbors in a more accurate manner. The baseline mechanism that we choose here is the Multi-dimensional Trust management framework (*mTrust*) that are discussed in our recent work [39]. In this work, we have shown that the (*mTrust*) framework outperforms other well-known security mechanisms for MANETs.

We use the following three parameters to evaluate the performance of the CAST framework: *Precision* (P), *Recall* (R), and *Communication Overhead* (CO). These parameters are defined as follows.

$$P = \frac{Num\ of\ Truly\ Malicious\ Devices\ Caught}{Total\ Num\ of\ Untrustworthy\ Devices\ Caught}$$

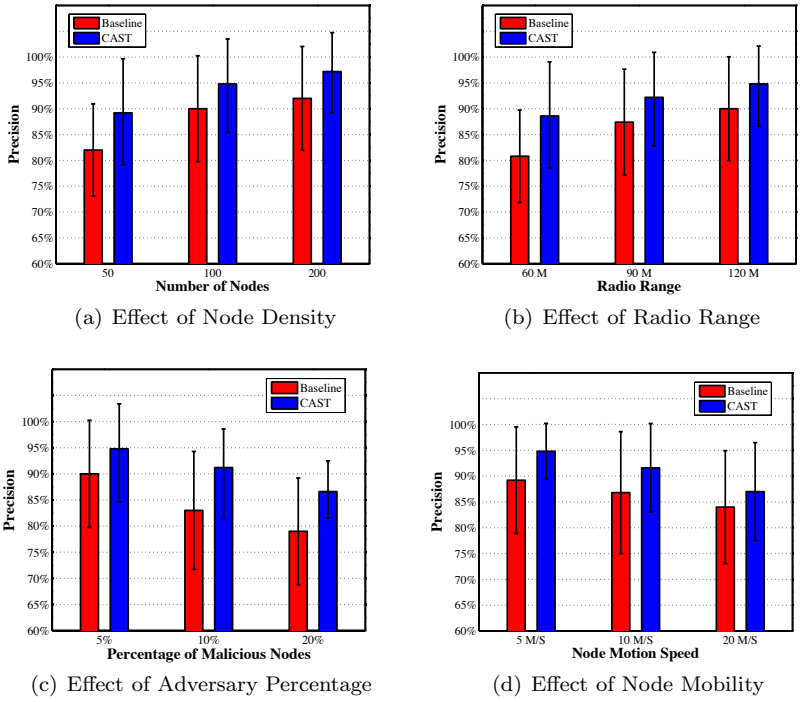
$$R = \frac{Num\ of\ Truly\ Malicious\ Devices\ Caught}{Total\ Num\ of\ Truly\ Malicious\ Devices}$$

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

Each simulation scenario has 20 runs with distinct random seeds, which ensures a unique initial node placement for each run. The experimental results are the average values over the 30 runs. The simulation results are shown in Figure 5, Figure 6, and Figure 7.

We find from Figure 5 and Figure 6 that the CAST framework generally outpeforms the mTrust scheme in terms of both precision and recall. More specifically, according to Figure 5(a) and Figure 6(a) both of them can yield a higher precision and recall value when the node density is higher. This is true because it is more likely to receive correct messages from others when there are a higher number of well-behaved mobile nodes. Figure 5(c) and Figure 6(c) tell us that both the precision and recall values decrease when there are a higher percentage of malicious nodes, which is pretty obvious, and the CAST framework can still yield high precision and recall values even when there are a lot of malicious nodes in MANET. We conclude from Figure 5(b) and Figure 6(b) that the precision and recall values for both CAST and mTrust will be degraded when the radio range is decreased. This is true because with a smaller radio range, it is more difficult for each node to obtain information from other nodes. Figure 5(d) and Figure 6(d) show that when the mobile

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.



(a) Effect of Node Density

(b) Effect of Radio Range

(c) Effect of Adversary Percentage

(d) Effect of Node Mobility

Fig. 5 Precision of *CAST* V.S. *mTrust*

nodes are traveling at a higher speed, it will be more challenging for both CAST and mTrust to detect the real adversaries.

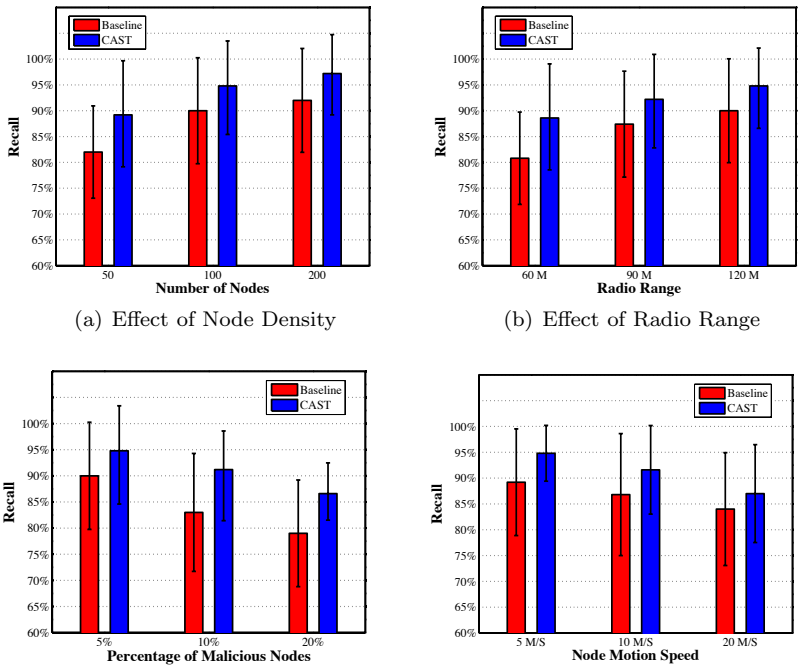
Figure 7 illustrates that the CAST framework normally has a lower communication overhead than the mTrust scheme, which will make it more desirable because of the restricted battery and bandwidth resources in MANETs.

5.2 Experiments on Android Phones

In addition to the simulation, we build an Android application which treat smartphones as components of a mobile network, and we have conducted experiments on the Android emulator. We use the device capabilities to collect sensor data and to perform reasoning over sensed data and contextual information using Jena. The experimental results are displayed in Figure 8 and Figure 9.

Figure 8(a) shows the environmental factors for an abnormal sensor report. According to the policy rule, this abnormal report is caused by the low temperature as well as the bad weather condition (heavy snow in this case). Therefore, we conclude that the abnormal report is caused by the environmental conditions, which is shown in Figure 8(b).

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.



(a) Effect of Node Density

(b) Effect of Radio Range

(c) Effect of Adversary Percentage

(d) Effect of Node Mobility

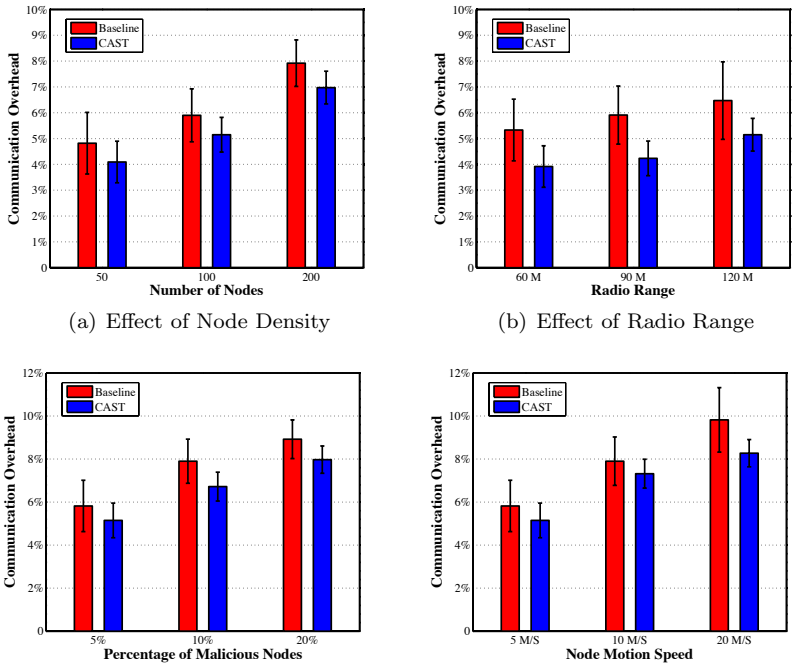
Fig. 6 Recall of *CAST* V.S. *mTrust*

On the other hand, Figure 9(a) illustrates the environment condition for another abnormal sensor report. According to this environment condition, the policy rule concludes that the abnormal report is NOT caused by the environment condition, which is displayed in Figure 9(b).

6 Conclusion

In this paper, a Context-Aware Security and Trust (CAST) framework is studied for Mobile Ad-hoc Networks to distinguish the truly malicious nodes from the malfunctioning nodes, both of which may exhibit misbehaviors. Through the use of various contextual information, such as channel status, speed, weather condition, and transmission signal strength, a node can determine the circumstance under which the misbehaviors occur. As a result, the node can then tell whether a node is forced to act as a misbehaving node or not, and reveal the truly malicious attackers. The simulation results as well as the experimental results on Android platform show that the CAST framework is highly resilient to malicious attackers, and it can accurately identify the malicious nodes from the faulty ones with a limited communication overhead.

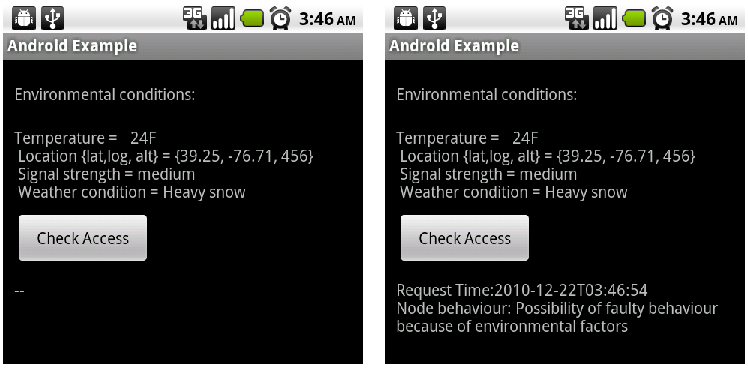
Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.



(a) Effect of Node Density (b) Effect of Radio Range

(c) Effect of Adversary Percentage (d) Effect of Node Mobility

Fig. 7 Communication Overhead of *CAST* V.S. *mTrust*



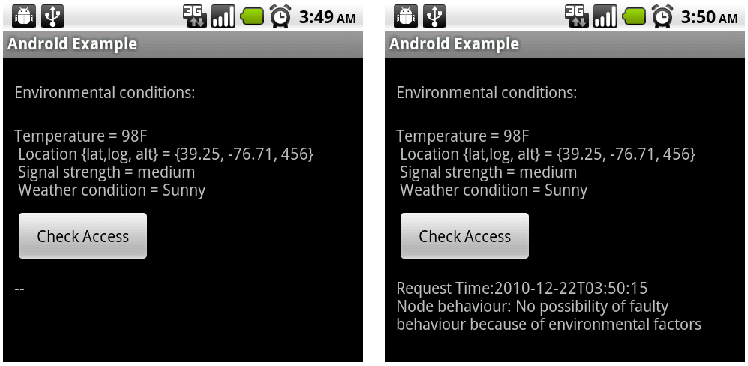
(a) Enviromental Factors in Faulty Case (b) Result of Policy Execution in Faulty Case

Fig. 8 Policy Execution in Faulty Case

References

1. 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.

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.



(a) Enviromental Factors in Normal Case

(b) Result of Policy Execution in Normal Case

Fig. 9 Policy Execution in Normal Case

2. 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.
3. 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.
4. 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.* IEEE Computer Society, May 2009, pp. 112–121.
5. 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. IEEE Computer Society, Aug. 2009, pp. 76–82.
6. W. Li, J. Parker, and A. Joshi, "Security through collaboration and trust in manets," *ACM/Springer Mobile Networks and Applications (MONET)*, vol. 3 num. 3, pp. 342–352, July 2012.
7. 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.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

13. 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.
14. 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.
15. L. Zhou and Z. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, no. 6, pp. 24–30, Nov/Dec 1999.
16. 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.
17. 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.
18. 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.
19. W. Li, A. Joshi, and T. Finin, "Atm: Automated trust management for mobile ad hoc networks using support vector machine," in *Mobile Data Management (MDM), 2011 12th IEEE International Conference on*, june 2011, pp. 291 –292.
20. W. Li, A. Joshi, and T. Finin, "Sat: an svm-based automated trust management system for mobile ad-hoc networks," in *2011 IEEE MILITARY COMMUNICATIONS CONFERENCE (MILCOM2011)*, nov. 2011, pp. 1102 –1107.
21. 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.
22. 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.
23. 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.
24. S. Buchegger and J.-Y. L. Boudec, "A robust reputation system for mobile ad-hoc networks," in *Proceedings of P2PEcon*, 2003.
25. 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.
26. 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.
27. 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.
28. Y. Ren and A. Boukerche, "Performance analysis of trust-based node evaluation schemes in wireless and mobile ad hoc networks," in *Proceedings of 2009 IEEE International Conference onCommunications, ICC '09.*, june 2009, pp. 1–5.
29. M. Sloman, "Policy driven management for distributed systems," *Journal of Network and Systems Management*, vol. 2, pp. 333–360, 1994.
30. L. Kagal, T. Finin, and A. Joshi, "A policy language for a pervasive computing environment," in *Proceedings of IEEE 4th International Workshop on Policies for Distributed Systems and Networks. POLICY 2003.*, 2003.
31. S. Godik and T. Moses, "Oasis extensible access control markup language (xacml)," 2002.

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.

32. B. Wu, J. Wu, E. B. Fernandez, M. Ilyas, and S. Magliveras, "Secure and efficient key management in mobile ad hoc networks," *Journal of Network and Computer Applications*, vol. 30, no. 3, pp. 937–954, 2007.
33. Google, "Android sensors," http://developer.android.com/reference/android/hardware/Sensor.html.
34. J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring," in *The Sixth Annual International conference on Mobile Systems, Applications and Services (MobiSys 2008)*, Breckenridge, U.S.A., June 2008.
35. J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson, "Jena: implementing the semantic web recommendations," in *Proceedings of the 13th international World Wide Web conference. WWW 2004*. New York, NY, USA: ACM, 2004, pp. 74–83.
36. F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics*, vol. 11, pp. 1–21, 1969.
37. G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ, USA: Princeton University Press, 1976.
38. 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.
39. W. Li, A. Joshi, and T. Finin, "Coping with node misbehaviors in ad hoc networks: A multi-dimensional trust management approach," in *Proceedings of the 11th International Conference on Mobile Data Management. MDM '10*. IEEE Computer Society, May 2010, pp. 112–121.

Author preprint: Wenjia Li, Anupam Joshi and Tim Finin, CAST: A Context-Aware Security and Trust Framework for Mobile Ad-hoc Networks Using Policies, Distributed and Parallel Databases, 10.1007/s10619-012-7113-3, Springer, to appear, 2013.