Location-aware, Trust-based Detection and Isolation of Compromised Nodes in Wireless Sensor Networks

Garth V. Crosby[1], Lance Hester[2], and Niki Pissinou[3]
(Corresponding author: Garth V. Crosby)

Department of Technology, College of Engineering, Southern Illinois University Carbondale[1],
Carbondale, IL 62966, USA
Honeywell International ACS- Wireless Group, Golden Valley, MN 55422[2]
Florida International University, College of Engineering[3],
Telecommunications and Information Technology Institute, Miami, FL 33174
(Email: crosby@engr.siu.edu, lance.hester@honeywell.com, pissinou@fiu.edu)

(Received Dec. 18, 2009; revised and accepted Feb. 17, 2010)

Abstract

The detection and isolation of compromised nodes in wireless sensor networks is a difficult task. However, failure to identify and isolate compromised nodes results in significant security breaches which lowers the integrity of gathered data. Using a reputation-based trust framework for wireless sensor networks we introduce a location-aware, trust-based protocol that detects and isolates compromised nodes. We employ a secure cluster formation algorithm to facilitate the establishment of trusted clusters via pre-distributed keys. Reputation and trust is built over time through the monitoring of neighboring nodes. Our scheme provides a mechanism for developing reputation and trust so that a device can determine whether other devices have been compromised, and take corrective action, through negative information sharing and independent trust-based decision making. We also present a simple location verification algorithm that utilizes the received signal strength information in the process of verifying reported location information. The effectiveness of our approach in the detection and isolation of compromise nodes is validated through simulation. The results indicate that our scheme provides an effective mechanism for detecting and isolating compromised colluding nodes.

Keywords: Beta reputation, detection, reputation, trust, wireless sensor networks

1 Introduction

Wireless sensor networks are emerging as very important tools in the gathering and dissemination of mission critical information in real time. As a result of the importance and usefulness of the data that will traverse these networks, it is necessary that sufficient security is in place to prevent the leakage or compromise of this data [5, 6, 35]. However, sensor networks pose unique new challenges that prevent the direct application of traditional security techniques [9, 19, 21, 34]. For economic viability, wireless sensor nodes are limited in power, computation capabilities and memory. The limitation of memory and processing capabilities makes public key cryptography and digital signature infeasible. In addition, the limited power of these tiny sensor nodes makes the communication overhead of traditional security algorithms unbearable. Furthermore, the lack of infrastructure, the insecure nature of wireless communication channel, and the hostile deployment environments present additional security vulnerabilities.

Several works to address the security issues of wireless sensor networks have been proposed. Perrig et al. [27] proposed a scheme for authenticated broadcast and data integrity. Karlof et al. [18] have identified a number of security vulnerabilities in wireless sensor routing protocols. Wood et al. [36] have comprehensively assessed denial of service (DoS) attacks on wireless sensor networks and suggested that wireless sensor networks' routing protocol must incorporate security features at the protocol design phase. The aforementioned works amongst others addressed the issue of external security attack in wireless sensor networks; however, only a few attempt to effectively detect and isolate compromised or malicious nodes that have gained access to the network through compromising encryption keys. Security techniques alone are not sufficient in securing these networks from internal attacks. It is necessary that trust-based mechanisms are employed since the compromised nodes would, by definition, have access to the network and possess the relevant keying material.

In this paper we present a localized protocol, which mitigates internal attacks through trust-based decision making. We employ location awareness, a common feature of many sensor network applications, and received signal strength in the validation of location information. Our reputation-based trust model is dynamic, that is, trust metrics are constantly being refreshed. Reputation in our work is a probabilistic distribution similar in nature as found in [10, 17]. Fundamental to our approach is the ability of nodes to monitor the traffic going in and out of their neighbors [10, 20]. We employ a data structure that stores the trust values in a trust table maintained by each node. Each node builds and maintains its trust table by monitoring its neighbors [20, 29]. The novelty lies in our approach of independent assessment and trust-based decision making on the reception of negative information and; the inclusion of location awareness as an element in trust building. Our contributions are:

- Resilience to node compromise- our scheme offers reputation-based monitoring that facilitates the detection and isolation of untrustworthy nodes.
- Location awareness- we integrate location awareness in our reputation-based scheme in order to enhance the integrity.

We also introduce a simple scheme that uses received signal strength to verify the location information.

2 Related Work

The issue of security in ad hoc and wireless sensor networks has been addressed in [3, 13, 16, 28, 33, 34, 36, 37]. In [36], the authors presented a comprehensive assessment of various denial-of- service (DoS) attacks and counter measures and how these apply to wireless sensor networks. These attacks are presented based on the security vulnerability of the physical, data link, network, and transport layers. In [18], the authors evaluated a number of wireless sensor network routing protocols and highlighted their weaknesses. They showed the security threats and proposed countermeasures.

A number of trust-based protocols for mobile ad hoc networks (MANETs) and wireless sensor networks have been proposed. We will first discuss ad hoc networks then highlight some works that specifically address wireless sensor networks. In [12], the authors proposed a secure routing solution to find an end-to-end route free of malicious nodes with the collaborative effort from the neighbors. Their solution also secures the network against colluding malicious nodes. A framework for computing and distributing trust in mobile ad hoc networks is also proposed. The proposed protocol is an extension of the Ad Hoc On-Demand Distance-Vector (AODV) Protocol and the authors' previous work Trusted-AODV (T-AODV) [11, 30]. In [38], the authors proposed a technique, Security-Aware ad hoc Routing (SAR), that incorporates security attributes as parameters into ad hoc route discovery. The protocol is an augmentation of AODV, which incorporates a trust level metric in the Route Request (RREQ) message. A hierarchical trust level is implemented among the nodes.

In [29], the authors proposed a framework for the establishment and management of trust in an ad hoc network without the aid of a central authority. A model is proposed in which trust is derived by assigning weights to various observable and measurable network activities or "trust categories". Examples of these include data packets received, control packets received, and data forwarded. The authors proposed an augmentation of the Dynamic Source Routing (DSR) protocol that uses passive acknowledgements with nodes operating in promiscuous mode to observe network communication. Each node is able to compute the trust level of other nodes. Assignment of trust levels to each node facilitates a trustworthy source to destination path. While the authors did not address wireless sensor networks specifically, we found this work to be helpful in the formulation of our cluster-based distributive trust framework for wireless sensor networks.

A secure trust-based public key authentication service to prevent nodes from receiving false public keys from malicious nodes is proposed in [23]. The system does not rely on any trusted-third party. The trust model follows the "web of trust" approach. The model uses digital signature as its form of introduction. Any node signs another's public key with its own private key to establish a web of trust. The nodes in the network monitor each other and update their trust table, which is stored at each node, accordingly. The public key management mechanism endures the false certificate issued by dishonest users and malicious nodes, and avoids them being selected as introducing nodes. The use of digital signature makes this approach impractical for sensor networks due to the limited power and computational capacity of wireless sensor nodes.

All of the trust-based papers discussed so far were developed for ad hoc networks and were not necessarily suitable for wireless sensor network due to the power, memory, and computational requirements of the nodes. We now discuss proposals that were specifically designed for wireless sensor networks.

A reputation-based framework for wireless sensor networks that utilizes Bayesian formulation and beta distribution is proposed in [10]. A watchdog mechanism resides in the middleware of each node and collects observable information. Second hand information is also included in the statistical computation of reputation. This information is gathered from nodes in the neighborhood. Direct observation and second hand information together facilitates a decentralized reputation-based system. The inclusion of second hand information would normally imply that the protocol is susceptible to badmouthing attacks. However, the authors remove this threat by allowing the nodes to only propagate good reputation information about other nodes. As the authors themselves point out, this resiliency comes at the cost of system efficiency as now the nodes cannot exchange their bad experiences about malicious/faulty nodes in the network. We use a similar trust model in our work. Each node maintains independent trust tables based on direct observation. We allow the sharing of only negative information. However, this is weighed against information obtained through direct interaction in any trust-based decision making. This approach efficiently deals with the threat of badmouthing attacks without any loss of system efficiency.

In [1], the authors proposed a "key infection" protocol that establishes a trust framework for the distribution of keys in a non-critical commodity sensor network. The nodes broadcast keying material as they are deployed and begin making contact with other nodes. Nodes gradually increase their broadcast transmission power until contact is made with another node. On contact, keying materials are exchanged in plain text with each other. One of the premises of this paper is that in a non-mission critical commodity wireless sensor network it is reasonable to assume that adversarial nodes are not present at the setup phase of the network. The authors argue that economic factors would prevent adversarial nodes from appearing during the setup phase since this would require the adversary to place many nodes in various locations with the hope that a network will be deployed in one of those locations. This would be a highly costly approach for the adversary and is considered impractical for non-mission critical commodity wireless sensor networks. We agree with the authors and incorporate their 'real world' attack model that excludes the existence of a global adversary during the network setup phase in our framework.

3 Modelling Reputation and Trust: A Probabilistic Model

Reputation and trust are the common basis of interactions that requires the performance of a future task based on past behavior. Trust and reputation have become important topics of research in many fields including psychology, philosophy, economics, and computer science. Expert researchers have employed various definitions of trust and reputations. We rely on the following definitions of these two terms:

- **Reputation:** perception that an agent creates about another agent's intention and norms, through direct or indirect observation of its' past actions [22].
- **Trust:** a subjective expectation an agent has about another's future behavior with respect to a specific action.

3.1 Basic Notation

In a wireless sensor network consisting of n nodes, we denote the set of all nodes as $S = \{s_1, s_2, \dots, s_n\}$ where $n \geq 2$. After deployment, pairs of nodes $\{s_i, s)j\} \subseteq S$ may interact directly with each other in order to perform a specific task that requires cooperation. Such an interaction may be considered successful by s_i if s_j cooperates in the performance of the task. From the perspective of s_i, the outcome of this interaction is successful if s_j cooperates or unsuccessful if s_j does not cooperate. The history of observed outcome between s_i and s_j, from the perspective of s_i, is recorded at any time t as a tuple, $H^t_{s_{ij}} = (C^t_{s_{ij}}, d^t_{s_{ij}})$ where the value of $c^t_{s_{ij}}$ is the number of successful interaction (cooperation) of s_j with s_i, while $d^t_{s_{ij}}$ is the number of unsuccessful interactions.

3.2 Beta Distribution

Various distributions such as beta, Poisson, and Gaussian have been used to represent the reputation of an agent (node). The beta distribution has been employed in a number of works [10, 17, 26]. Jϕsang [17], in particular, has provided a thorough treatment of beta distribution and its usefulness in reputation systems. We opted to use beta distribution because of its simplicity, strong foundation on statistical theory, the fact that its computation requires mainly two parameters (which make it quite applicable for the memory constrained wireless sensor nodes), and its appropriateness in representing the probability distribution of binary events.

We will now define the beta distribution, its probability density function, and its statistical expectation. The beta distribution function $B(v, w)$ is defined as:

$$B(v,w) \quad = \quad \int_0^1 u^{v-1}(1-u)^{\omega-1}du,$$

where the shape parameters $v > 0$ and $\omega > o$. The beta probability density function (pdf) is defined as:

$$f(p|v,\omega) \quad = \quad p^{v-1}(1-p)^{\omega-1}/B(v,\omega),$$

where $0 \leq p \leq$ and $v, \omega > 0$. For mathematical convenience, the beta probability density function $f(p|v,\omega)$ can be expressed using the gamma function τ as:

$$f(p|v,\omega) \quad = \quad \frac{\tau(v+\omega)}{\tau(v)\tau(\omega)}p^{v-1}(1-p)^{\omega-1},$$

where $0 \leq p \leq 1$, $v > 0$, $\omega > 0$ with the restriction that the probability variable $p \neq 0$ if and $v < 1$ if The expected value for the beta distribution is defined as:

$$E(p) \quad = \quad v/(v+\omega),$$

where p is a probability variable.

3.3 Modelling Reputation

The reputation of node s_j that is maintained at node s_i at any time t is defined as [17]:

$$R^t_{s_{ij}} \quad = \quad \frac{\tau(v+\omega)}{\tau(v)\tau(\omega)}p^{v}(1-p)^{\omega},$$

where $0 \leq p \leq 1$, $v > 0$, $\omega > 0$; setting $v^t_{s_{ij}} + 1$ and $\omega = d^t_{s_{ij}} + 1$, where $c^t_{s_{ij}}, d^t_{s_{ij}} \geq 0$. A simple check to assess the accuracy of our model is to compute the reputation between two nodes with no prior experience, that is, $c^t_{s_{ij}}$, $d^t_{s_{ij}} = 0$.

This leads to $R^t_{s_{ij-}} = Beta(1,1) = uni(0,1)$. This agrees with intuition that without prior information, the most reasonable reputation function is the uniform distribution, which means that both possibilities are equally likely of occurring in the future.

3.4 Modelling Trust

We have employed the beta distribution function in modelling reputation between two nodes; however, equally important is the requirement to have a means of comparing the relative trustworthiness of the nodes within the context of the network. Consistent with our definition of trust, we define a trust metric that quantifies the level of trust the nodes are willing to exhibit towards each other based on past experiences. We define our trust metric between two nodes s_i and s_j, from the perspective of s_i, as [17]:

$$\begin{aligned} T_{s_{ij}} &= E(R^t_{s_{ij}}) = v/(v+\omega) \\ &= c^t_{s_{ij}} + 1/c^t_{s_{ij}} + d^t_{s_{ij}} + 2. \end{aligned}$$

This gives a trust metric in the range [0, 1] where the value 0.5 represents a neutral rating.

3.5 Modelling Uncertainty

Based on our trust metric it is easy to see that various nodes can exhibit the same level of trust, though, with a varying number of interactions between the two parties. For example, let us consider the trust levels of nodes s_j and s_k, from the perspective of s_i. The trust values are:

$$\begin{aligned} T_{s_{ij}} &= 0.5 \text{ where } c^t_{s_{ij}} = 2 \text{ and } d^t_{s_{ij}} = 2; \\ T_{s_{ij}} &= 0.5 \text{ where } c^t_{s_{ij}} = 21 \text{ and } d^t_{s_{ij}} = 21. \end{aligned}$$

A pertinent question is: Which trust level would s_i be willing to exercise a greater belief in? The intuitive and correct response should be $T_{s_{ik}}$ based on the fact that s_i has a greater number of observations to assess the trust level of s_k. In other words, a greater amount of information should lead to a greater level of certainty. It is generally accepted that uncertainty arises from the lack of complete information. To somewhat compensate for uncertainty and to better facilitate trust-based decision making we introduce a confidence metric derived from the standard definition of entropy.

For a continuous probability density function (pdf), $f(x)$, information content (or entropy) is defined in the literature as:

$$I = -\int_{+\infty}^{-\infty} f(x)\ln(f(x))dx.$$

We adhere to the school of thought that views entropy as a measure of uncertainty about the occurrence of a future event. In light of this, we define our confidence metric between two nodes s_i and s_j, from the perspective of s_i, as:

$$n_{ij} = |\frac{1}{-\int_{+\infty}^{-\infty} f(x)\ln(f(x))dx}|,$$

where $-\int_{+\infty}^{-\infty} f(x)\ln(f(x))dx \neq 0$, in this particular case, represents the beta density function which models the reputation between the two nodes.

3.6 Updating Reputation

Given the reputation, $R^t_{s_{ij}}$, between two nodes s_i and s_j, the reputation q time later, $R^{(t+q)}_{s_{ij}}$, where $q > 0$, is computed as by incorporating the number of successful interactions $(c^Q_{s_{ij}})$ and the number of unsuccessful interactions $(d^Q_{s_{ij}})$ during the period t to $(t+q)$ as follows :

$$\begin{aligned} c^{t+q}_{s_{ij}} &= c^t_{s_{ij}} + c^Q_{s_{ij}}; d^{t+q}_{s_{ij}} = d^t_{s_{ij}} + d^Q_{s_{ij}}; \\ R^{(t+q)}_{s_{ij}} &= Beta(c^{t+q}_{s_{ij}} + 1, d^{t+q}_{s_{ij}} + 1). \end{aligned}$$

3.7 Aging

Intuitively, recently obtained information used to approximate trustworthiness should carry a greater weight than older information. This prevents compromised or malicious nodes from taking advantage of initial high trustworthiness. We achieved this through exponential averaging as proposed in [17] and employed in [10].

$$\begin{aligned} c^{new}_{s_{ij}} &= (w_{age} * c^t_{s_{ij}}) + m; \\ d^{new}_{s_{ij}} &= (w_{age} * d^t_{s_{ij}}) + n. \end{aligned}$$

Here w_{age} is the age weighting factor and takes values in the range (0, 1); m and n are the number of successful interactions and unsuccessful interactions respectively, for the period from the last computation of $c^t_{s_{ij}}$ and $d^t_{s_{ij}}$ to the current time.

4 Trust Based Protocol

In this section, we highlight our trust-based framework and state assumptions, the threat model, and describe the details of our algorithms.

4.1 Assumptions of Model

A number of assumptions are made concerning the framework in which the wireless sensor nodes operate. First, a reliable link layer protocol is assumed. Once the clusters are formed they maintain the same members, except for cases where nodes are blacklisted, die, or when new nodes join the network. All the nodes communicate via a shared bidirectional wireless channel and operate in the promiscuous mode. The nodes remain fairly stationary most of the time. The nodes know their locations and have unique local IDs. They are able to determine the received signal strength of all the received packets. In addition, each wireless node maintains a data structure that facilitates the storage of a trust table that includes all their one-hop neighbors.

We do not consider key distribution, but we assume that each node has three keys; a master, cluster, and pairwise. The master key is shared by every node and facilitates broadcast by the base station. Members of each cluster share the cluster key. Each cluster has a different cluster key. This key facilitates multicasting communication from the base station to a cluster and also group communication within the clusters themselves. The pairwise key allows node-to-node communication. The master key, we assume, is placed in the nodes during the manufacturing process. The manufacturing process cannot be compromised.

4.2 Threat Model

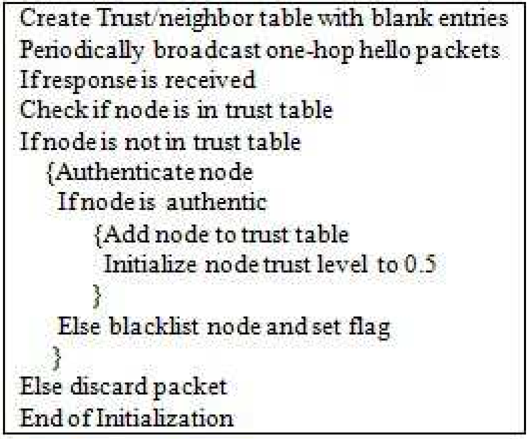
We consider a threat scenario which consists of malicious (or compromised) nodes that are deployed after the setup phase of the network. Attackers have the ability to collude.

We consider a threat scenario which consists of malicious (or compromised) nodes that are deployed after the setup phase of the network. Attackers have the ability to collude.We consider a threat scenario which consists of malicious (or compromised) nodes that are deployed after the setup phase of the network. Attackers have the ability to collude.

4.3 Location-aware Compromised Node Detection and Isolation

Within the context of the modelling framework we have established for reputation, trust, and uncertainty, we now describe our location-aware reputation-based trust mechanism. Our system has three phases: node discovery & trust initialization, maintenance, and revocation. The node discovery & trust initialization occurs first from a chronological perspective while the maintenance and revocation phase overlap in time.

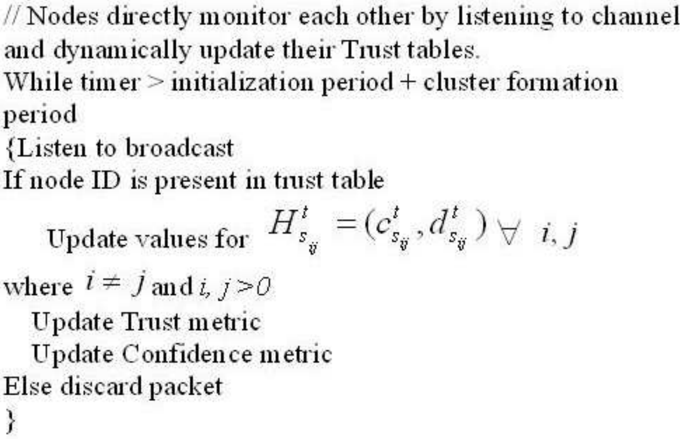
In the node discovery & initialization phase, which immediately follows deployment, each node periodically, in the order of seconds, broadcasts one-hop hello packets to discover its neighbors. On the reception of a hello message from node s_i, node s_j replies with an authenticated message using the pairwise key. Embedded in the reply are s_j's node ID and location information. If node s_j is verified to be authentic, then it is recorded in 's neighbors list (trust table), and its trust value is initialized. Figure 1 gives a high level description of the node discovery & trust initialization phase algorithm. Immediately following the node discovery & trust initialization phase, a secure cluster formation algorithm is executed. The detail of this is explained later.



```
Create Trust/neighbor table with blank entries
Periodically broadcast one-hop hello packets
If response is received
Check if node is in trust table
If node is not in trust table
   {Authenticate node
    If node is authentic
        {Add node to trust table
         Initialize node trust level to 0.5
        }
    Else blacklist node and set flag
   }
Else discard packet
End of Initialization
```

Figure 1: High level description of the node discovery and trust initialization algorithm

The maintenance phase involves updating reputation, trust, and confidence metrics according to the modelling parameters we described in Section 3. This phase occurs a certain time after the node discovery & trust initialization phase and cluster formation period. During this phase the nodes monitor the traffic coming in and out of their neighbors. Periodically, the history of observed outcome is updated by updating $H^t_{s_{ij}} = (c^t_{s_{ij}}, d^t_{s_{ij}})$. For each of these updates, the corresponding trust and confidence metrics are also updated. The trust metric is updated by computing $T_{s_{ij}} = E(R^t_{sij})$. The confidence metric is updated by computing using Equations 1. Figure 2 gives a high level description of the maintenance phase algorithm.



```
// Nodes directly monitor each other by listening to channel
and dynamically update their Trust tables.
While timer > initialization period + cluster formation
period
{Listen to broadcast
If node ID is present in trust table
     Update values for $H^t_{s_{ij}} = (c^t_{s_{ij}}, d^t_{s_{ij}}) \forall\ i, j$
where $i \neq j$ and $i, j > 0$
   Update Trust metric
   Update Confidence metric
Else discard packet
}
```

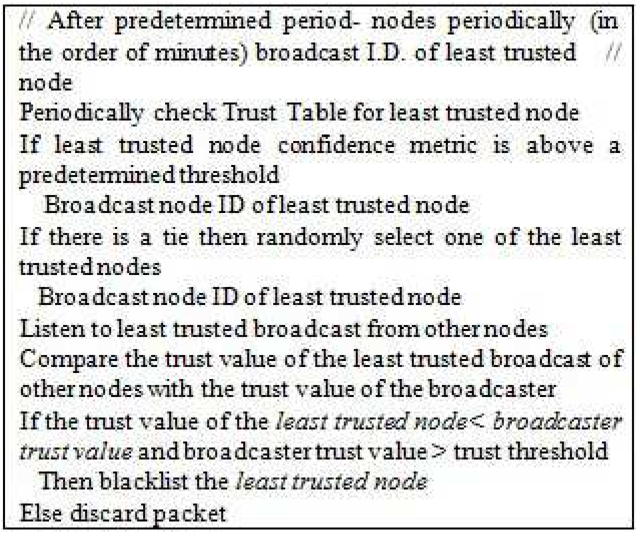
Figure 2: High level description of maintenance phase algorithm

The purpose of our revocation algorithm is to remove untrustworthy nodes from the network. Each node periodically checks its trust table for the node that has the lowest trust metric. If the node selected also has a confidence value above a predetermined threshold then that node is blacklisted. In addition, its node ID is broadcasted as being *untrustworthy*. (Note that the confidence metric value is a measure of the certainty of the calculated trust value. The higher the confidence metric, the more certain the calculated trust value). If in selecting the node with lowest trust metric there is a tie with both the trust and confidence values, then one of the nodes is randomly selected. On the reception of an *untrustworthy* node broadcast, each node searches its trust table for a match. The broadcasted untrustworthy node is blacklisted by other nodes by setting its trust value to -1 if all of the following conditions are met:

1) the broadcasted untrustworthy node has a lower trust value than the broadcaster (sender of the untrustworthy node's ID), and

2) the broadcaster trust level is above a certain threshold.

No future trust update is performed or cooperation facilitated between nodes with -1 trust values. This simple algorithm is in agreement with social exchanges, among a group of people, which penalize others based on third party recommendations. Within this context, penalty is based on the relative trustworthiness of the third party making the recommendation and the subject of the recommendation. Figure 3 describes our revocation phase algorithm.



```
// After predetermined period- nodes periodically (in the order of minutes) broadcast I.D. of least trusted // node
Periodically check Trust Table for least trusted node
If least trusted node confidence metric is above a predetermined threshold
   Broadcast node ID of least trusted node
If there is a tie then randomly select one of the least trusted nodes
   Broadcast node ID of least trusted node
Listen to least trusted broadcast from other nodes
Compare the trust value of the least trusted broadcast of other nodes with the trust value of the broadcaster
If the trust value of the least trusted node< broadcaster trust value and broadcaster trust value > trust threshold
   Then blacklist the least trusted node
Else discard packet
```

Figure 3: High level description of revocation phase algorithm

4.4 Location Verification Algorithm

We assume that our nodes are aware of their location. To achieve this, generally two approaches are employed: range- free such as [2, 14, 15, 24] and range-based such as [4, 8, 25, 32]. While we recognize the advantages and disadvantages of the two approaches, we believe that any of these, suitably adapted, can be employed in our model. However, for convenience, we assume that each node is able to determine its precise location.

Sastry et ta. [31] proposed the Echo protocol, a secure location verification algorithm. However, their method does not verify the precise location but rather if a node is within the region it claimed. In addition, the nodes must be able to communicate using both radio frequency and sound (typically ultrasound frequencies). Our protocol differs in that we further limit the location possibilities to a narrow region along a concentric circle. In addition to location awareness, we assume that a distribution of received signal strength variation is known a prior, and that it is uniform. Further to this we assume that the standard deviation of the distribution has been computed beforehand. Later we will explain the reasons for the assumptions of the standard deviation and the known distribution of the received signal. Finally, all parameters relating to the path loss model are assumed known.

Our protocol aims to validate location information rather than to detect nodes that deliberately report falsified location information. However, through our validation process, nodes that falsify location information can be detected and isolated. Our protocol is described as follows: node s_i, wishing to obtain the location of another, node s_j, generates a random nonce and sends to node s_j. Node s_j responds by sending a message that includes its node ID, location (XY_j), and a message authentication tag encrypted with the pairwise key, L_{ji}, as shown in Figure 4. On reception of the reply, node s_i decrypts and authenticates the message by computing the message authentication code (MAC). If the response is authentic and fresh, then s_i node computes node s_j's approximate distance by using the received signal strength of the reply from node s_j.

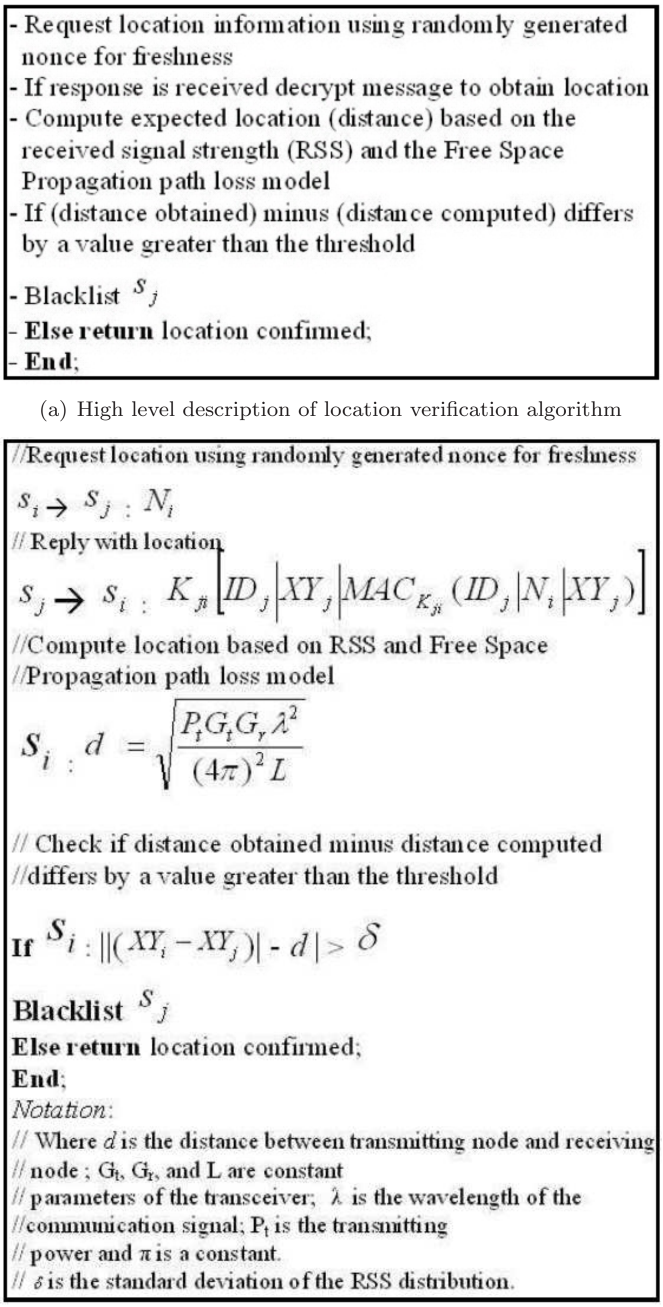
If the difference between the reported and computed locations is greater than the threshold (i.e. one or more standard deviation), then node s_i blacklists node s_j, otherwise node s_j's reported location is confirmed.

4.5 Secure Cluster Formation Algorithm

We use the protocol proposed in [7] to establish trusted clusters in the initial stages of the network through the use of pre-distributed keys. Immediately after the Node Discovery & Trust Initialization Phase, each node s_i, waits a random time (according to an exponential function) before broadcasting, at most, one "Cluster head" message to its neighbors declaring its decision to become a cluster head. This broadcast is encrypted using K_m so that the message is transmitted securely in an untrustworthy environment. Only nodes preloaded with the master key will be able to decrypt the message. As in [7], the encrypted message contains the ID of the node i (ID_i), its cluster key K_c^i, and an authentication tag as follows:

$$E_{K_m}(ID_i|K_c^i|MAC_{K_m}(\langle ID_i|K_c^i\rangle)).$$

Upon receiving a "Cluster head" message, a node de-



```
- Request location information using randomly generated
  nonce for freshness
- If response is received decrypt message to obtain location
- Compute expected location (distance) based on the
  received signal strength (RSS) and the Free Space
  Propagation path loss model
- If (distance obtained) minus (distance computed) differs
  by a value greater than the threshold
- Blacklist $s_j$
- Else return location confirmed;
- End;
```

(a) High level description of location verification algorithm

```
//Request location using randomly generated nonce for freshness
$s_i \rightarrow s_j : N_i$
// Reply with location
$s_j \rightarrow s_i : K_{ji}\left[ID_j | XY_j | MAC_{K_{ji}}(ID_j | N_i | XY_j)\right]$
//Compute location based on RSS and Free Space
//Propagation path loss model
$S_i : d = \sqrt{\frac{P_t G_t G_r \lambda^2}{(4\pi)^2 L}}$
// Check if distance obtained minus distance computed
//differs by a value greater than the threshold
If $S_i : ||(XY_i - XY_j)| - d| > \delta$
Blacklist $s_j$
Else return location confirmed;
End;
Notation:
// Where d is the distance between transmitting node and receiving
// node ; G_t, G_r, and L are constant
// parameters of the transceiver; λ is the wavelength of the
//communication signal; P_t is the transmitting
// power and π is a constant.
// δ is the standard deviation of the RSS distribution.
```

(b) Retailed description of location verification algorithm

Figure 4: The location verification algorithms

crypts and authenticates the message. It then does one of the following:

1) If the node has not made any decision about its role as yet and if the "Cluster head" is sent by a node whose trust level is above the threshold, it joins the cluster of the node that sent the "Cluster head" and cancels its timer. It joins the cluster by replying to the cluster head with a message encrypted by K_c^i and containing its ID_i;

2) If the node has already decided its role, it rejects the message. This happens because the node already joined a cluster or the node already sent a "Cluster head" message;

3) If the "Cluster head" message is from a previously blacklisted node, the message is ignored.

If after a period of time a node did not receive a "Cluster head" message or no node responded to its "Cluster head" message, then it declares itself a cluster head. In this case, the cluster head has only itself as a member of the cluster.

5 Analysis of the Accuracy of the Location Verification Scheme

Figure 5 gives a depiction of the region in which a location claim can be verified by node s_i to be correct. Node s_i cannot verify the claim of any node, such as node s_j, which is located beyond the shaded area. If the distance reported exactly matches what was calculated, then the node whose location is being verified would lie at any point on the circumference of the inner circle. This occurrence, however, would not be expected to occur frequently due to the variation in the received signal strength. Our challenges here are:

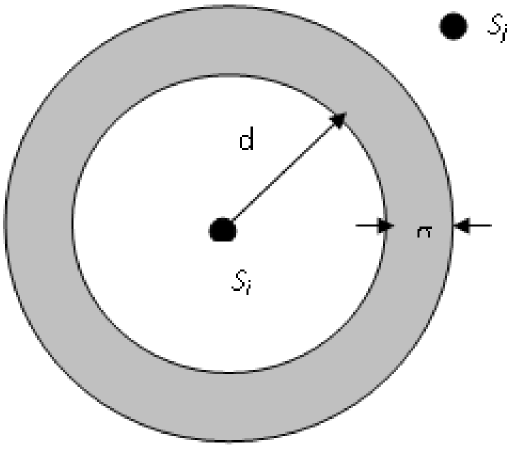


Figure 5: Depiction of location verification region

1) to ensure that the shaded region is narrow so that the mechanism is precise, that is, $n\sigma$ is small where $n \geq 1$, and

2) to ensure that the region within the diameter of $d + n\sigma$ includes almost all of the possible locations of a truthful node. If the former condition is not met, then the mechanism will allow the scope for false reporting without penalty. Failure to meet the latter condition will result in a significant percentage of nodes being wrongly penalized for authentic reports of location. Selecting the correct multiple of σ for the application is important in properly balancing the requirements of these two important conditions.

Figure 6 shows a uniform distribution and the percentage of data points corresponding to multiples of the standard deviation. From Table 1 it can be deduced that if one standard deviation is used as our threshold, then it is expected that 84.14% of the possible locations points would be covered by the circle with diameter $d + n\sigma$, allowing us an error margin of 15.86%. If two standard deviations are used, then 97.73% of the locations are covered with error margin of 2.27%. The accuracy approaches 100% as we increase the multiple of standard deviation.

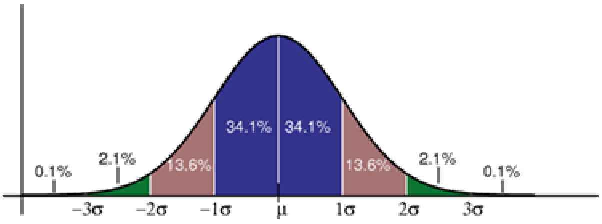


Figure 6: Uniform distributed data sample of RSS

If we assume that: $d >> \sigma$, and that σ is relatively small, then we can safely use 3 or more σ so that our accuracy is greater than 99% while maintaining a very narrow region of possible locations. This implies that the shaded region in Figure 5, in relation to the distance d, would be much narrower than shown. It also demonstrates the precision of our approach in verifying location, bearing in mind that we are solely interested in distance verification. The particular orientation of the nodes is not relevant to this discussion.

The security of our mechanism lies in the strength of the encryption algorithm. Of course a node can always adjust its power level and falsely report its location. However, we do not attempt to prevent nor detect these types of scenarios; we are only concerned in using signal strength to validate location information and thus aid in the prediction of future cooperation.

6 Simulations

We primarily use a discrete event C/C++ simulator to evaluate our protocol. We randomly distribute the nodes in a 50 m^2 area. For convenience, we set the aging factor to 1. We deployed nodes numbering in the range of 5 and 100 for each set of experiments. We have defined four types of nodes: good, bad, colluding, and random. Good nodes are set to have a packet drop rate less than 20 %. Bad nodes have a packet drop rate of 70% or greater. Colluding nodes are bad nodes with a packet drop rate of less than 20% amongst themselves, and greater than 70% with other non-colluding nodes, whether these non-colluding nodes are good or bad. We use a simple message exchange protocol where each node randomly sends messages to any nodes in its neighbor list that is not blacklisted. We investigate only intra-cluster communication and the efficacy of our protocol in isolating malicious or compromised members. For the simulation, we assume that all the cluster members share a common cluster key and thus can overhear the communication within the cluster. We set the transceiver range such that all nodes are within broadcast range. We use the mean of 30 simulation runs for each experiment.

Figures 7 and 8 demonstrate the average trust and average drop packet ratio respectively, versus the simulation round for a good node. The total number of nodes in the cluster is 20; this includes two bad nodes and two random nodes that we arbitrarily selected.

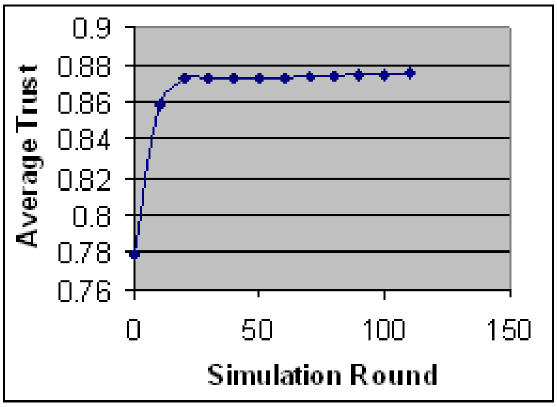


Figure 7: Average trust versus simulation round

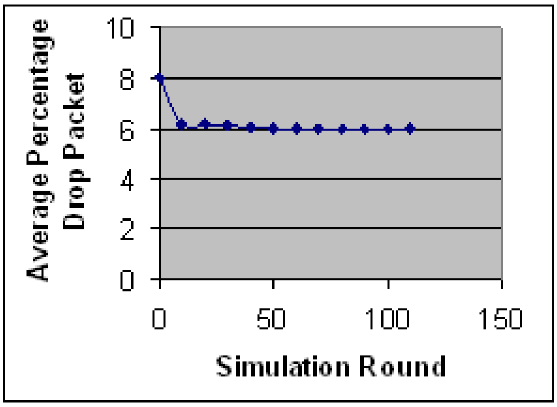


Figure 8: Average percentage drop packet versus simulation round

Table 1: Confidence intervals of a uniform distributed sample of RSS

Multiple of standard deviations	Percentage of sample points included	Percentage of all possible location covered by circle with diameter $(d + n\sigma)$
σ	68.27%	84.14%
2σ	94.45%	97.73%
3σ	99.73%	99.87%
4σ	99.99%	99.99%

No colluding nodes are employed in this set of experiments. Figure 7 shows that at the outset of the simulation, the average trust increases linearly however, after a period of time (simulation round) the average trust remains constant. This occurs because after a certain number of simulation rounds, increased cooperation among nodes makes less impact on the computed trust levels. The trust level stabilizes to the average cooperation being experienced with its neighbors. Figure 8 shows that the average percentage packet drop remains fairly constant after a limited number of simulations. This is because the bad nodes have been identified and blacklisted; therefore less packets are dropped by neighboring nodes (cluster members). Early detection and isolation allow savings in bandwidth and communication power that would be needed for the re-transmission of packets.

Figure 9 demonstrates the relationship between the Average Percentage Drop Packet and the Average Trust for the neighbors of a bad node. Initially, we see that the average gradually increases, and then rapidly increases to almost 100%. We explain this trend as follows: at the outset, the nodes cooperate with the compromised node; however, as its neighbors record its activities and compute its trust level, they initiate a process of blacklisting, this takes a brief period before basically all the nodes in the cluster blacklist this node.

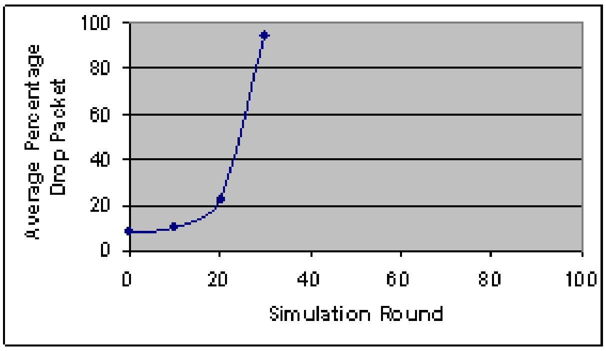


Figure 9: Average percentage drop packet versus simulation round of bad node

We set up a different experiment to study the ability of the protocol to isolate compromise nodes. We set 5% of the nodes as bad while the others are good. No colluding or random nodes are included. We systematically increase the number of nodes in increments of 5 from 0 to 60. We use the same data exchange protocol as before, and the data rate was set to 2 Mb/s. Packet lengths are 10kbit and one is generated every second. Figure 10 shows that the probability of compromised node detection is certain when the number of neighboring nodes is 15 or less. As the number of neighboring nodes increases, the probability of blacklisting by more than 40% of the neighboring nodes decreases. This is as a result of the increasing collision of packets as the density of the cluster increases. This collision causes an increase in false positives by the monitoring nodes.

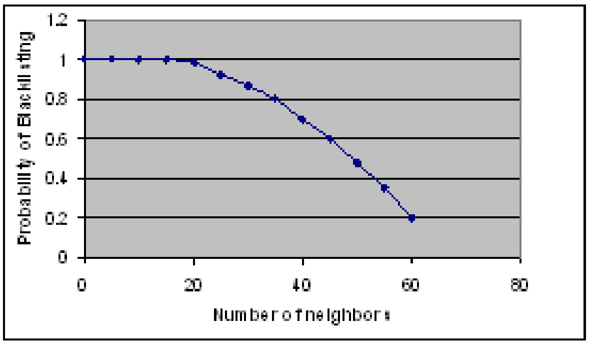


Figure 10: Probability of compromised node isolation by more than 40% of neighbors

We repeat the previous experiment setup with the following modifications: 100 nodes were deployed; only colluding and good nodes were included. We systematically increased the percentage of colluding nodes to a maximum of 60%. We determined the probability of isolating an arbitrarily selected colluding node by 30% of its neighbors (non-colluding neighbors) as the percentage of colluding nodes increases. Figure 11 shows that this probability is 1 for a relatively small percentage of colluding nodes. However; it decreases exponentially and approaches 0 with further increase in the percentage of colluding nodes. This occurs because it becomes increasingly difficult for a good node to spread negative information that results in the isolation of compromised colluding nodes in the presence of increasing percentage of colluding nodes. This is as a result of the accumulation of falsified reporting by the colluding nodes.

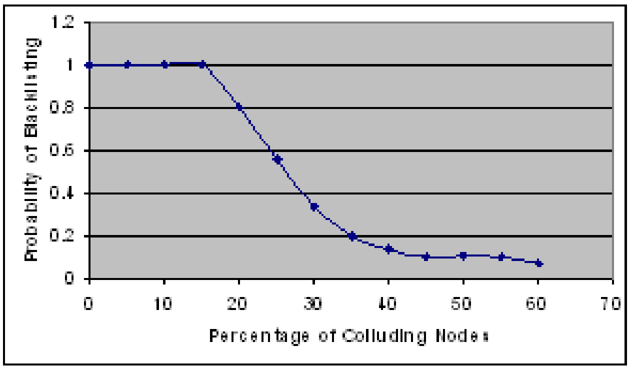


Figure 11: Probability of blacklisting colluding nodes by 40% of neighbors

7 Conclusion

We have presented a location-aware trust-based localized protocol that is able to detect and isolate compromised or malicious nodes. Our protocol is designed in the context of a cluster-based network model with nodes that have unique local IDs. We employ a beta reputation-based trust model. We introduce a simple location verification protocol that validates reported location information. Our protocol is assessed by its ability to detect and isolate compromised nodes. Simulations indicate that our protocol effectively detects and isolates compromised nodes even in the presence of colluding nodes.

References

[1] R. Anderson, H. Chan, and A. Perrig, “Key infection: Smart trust for smart dust,” *12th IEEE International Conference on Network Protocols (ICNP’04)*, pp. 206-215, Berlin, Germany, Oct. 2004.

[2] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less low cost outdoor localization for very small devices,” *IEEE Personal Communication Magazine*, vol. 7, pp. 28-34, Oct. 2000.

[3] E. Callaway, “Secure low-power operation of wireless sensor networks,” *Sensors Online*, vol. 21, no. 1, 2004.

[4] X. Cheng, A. Thaeler, G. Xue, and D. Chen, “TPS: a time-based positioning scheme for outdoor wireless sensor networks,” *IEEE INFOCOM ’05*, Miami, FL, Mar. 2005.

[5] A. K. Das, “An identity-based random key predistribution scheme for direct key establishment to prevent attacks in wireless sensor networks,” *International Journal of Network Security*, vol. 6, no. 2, pp. 134-144, 2008.

[6] A. K. Das, “ECPKS: An improved location-aware key management scheme in static sensor networks,” *International Journal of Network Security*, vol. 7, no. 3, pp. 358-369, 2008.

[7] T. Dimitriou, and I. Krontiris, “A localized, distributed protocol for secure information exchange in sensor networks,” *19th IEEE International Parallel and Distributed Processing Symposium (IPDPS’05)*, pp. 240, Denver, Colorado, 2005.

[8] L. Doherty, K. S. Pister, and L. E. Ghaoui, “Convex optimization methods for sensor node estimation,” *IEEE INFOCOM ’01*, Anchorage, Alaska, Apr. 2001.

[9] F. Dressler, “Authenticated reliable and semi-reliable communication in wireless sensor networks,” *International Journal of Network Security*, vol. 7, no. 1, pp. 61-68, 2008.

[10] S. Ganeriwal, and M. B. Srivastava, “Reputation-based framework for high integrity sensor networks,” *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN ’04)*, pp. 66-77, Washington, D.C., USA, Oct. 25, 2004.

[11] T. Ghosh, N. Pissinou, and K. Makki, “Collaborative trust-based secure routing against colluding malicious nodes in multi-hop ad hoc networks,” *Annual Conference on Local Computer Networks (LCN)*, pp. 224-231, Tampa, USA, 2004.

[12] T. Ghosh, N. Pissinou, and K. Makki, “Towards designing a trusted routing solution in Mobile Ad Hoc networks,” *ACM Journal, Mobile Networks and Applications (MONET)*, Special issue on Non-Cooperative Wireless Networking and Computing, 2005.

[13] C. Giovanni, “Topology for Denial of Service,” *Endeavour Systems Inc, White Paper*, July 12, 2000.

[14] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher, “Range-free localization scheme in large scale sensor networks,” *ACM MOBICOM ’03*, San Diego, Ca, Sep. 2003.

[15] L. Hu, and D. Evans, “Localization for mobile sensor networks,” *ACM MOBICOM ’04*, Philadelphia, PA, Sep. 2004.

[16] K. Jones, A. Wadaa, S. Olariu, L.Wilson, and M.Eltoweissy, “Towards a New Paradigm for Securing Wireless Sensor Networks,” *ACM New Security Paradigms Workshop*, pp. 115-121, Ascona, Switzerland, 2003.

[17] A. Jϕsang, and R. Ismail, “The beta reputation system,” *the 15th Bled Electronic Commerce Conference*, pp. 33-48, Bled, Slovenia, 2002.

[18] C. Karlof, and D. Wagner, “Secure routing in wireless sensor networks: Attacks and countermeasures,” *First IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 113-127, 2003.

[19] T. Landstra, S. Jagannathan, and M. Zawodniok, “Energy-efficient hybrid key management protocol for wireless sensor networks,” *International Journal of Network Security*, vol. 9, no. 2, pp. 121-134, 2009.

[20] S. Marti, T. J. Giuli, K. Lai, and M. Baker, “Mitigating Routing Misbehavior In Mobile Ad Hoc Networks,” *6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 255-

265, Boston, Massachusetts, United States, Aug. 2000.

[21] A. Mohaisen, D. Nyang, and K. Lee, "Hierarchical grid-based pairwise key pre-distribution in wireless sensor networks," *International Journal of Network Security*, vol. 8, no. 3, pp. 282-292, 2009.

[22] L. Mui, M. Mohtashemi, and A. Halberstadt, "A Computational Model of Trust and Reputation," *35th Annual Hawaii International Conference on System Sciences (HICSS-35'02)*, pp. 188, Hawaii, 2002.

[23] E. C. H. Ngai, and M. R. Lyu, "Trust and Clustering Based Authentication Services in Mobile Ad Hoc Networks," *24th International Conference on Distributed Computing Systems Workshops (ICDCSW'04)*, Tokyo, Japan, Mar. 2004.

[24] D. Niculescu, and B. Nath, "DV based Positioning in Ad Hoc Networks," *Journal of Telecommunication Systems*, 2003.

[25] D. Niculescu, and B. Nath, "Ad Hoc Positioning System (APS) using AoA," *IEEE INFOCOM '03*, San Francisco, CA, Apr. 2003.

[26] J. Patel, W. T. L. Teacy, N. R. Jennings, and M. Luck, "A Probabilistic Trust Model for Handling Inaccurate Reputation Sources," *the Third International Conference on Trust Management*, pp. 193-209, Rocquencourt, France, 2005.

[27] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J.Tygar, "SPINS: Security protocols for sensor networks," *Seventh Annual ACM International Conference on Mobile Networking and Computing (Mobicom 2001)*, pp. 189-199, Rome Italy, 2001.

[28] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 53-57, 2004.

[29] A. A. Pirzada, and C. McDonald, "Establishing trust in pure Ad-hoc networks," *The 27th Australasian Computer Science Conference*, Dunedin, New Zealand, 2004.

[30] N. Pissinou, T. Ghosh, and K. Makki, "Collaborative trust-based secure routing in multihop Ad Hoc networks," *The Third IFIP-TC6 Networking Conference (Networking '04)*, Athens, Greece, 2004.

[31] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," *the Second ACM Workshop on Wireless Security (WiSe)*, San Diego, California, Sep. 2003.

[32] A. Savvides, C. Han, and M. Srivastava, "Dynamic fine-grained localization in Ad-hoc networks of sensors," *ACM MOBICOM*, Rome, Italy, July 2001.

[33] B. Sieka and A. D. Kshemkalyani, "Establishing authenticated channels and secure identifiers in Ad-hoc networks," *International Journal of Network Security*, vol. 5, pp. 55-61, 2007.

[34] S. Slijepcevic, M. Potkonjak, V. Tsiatsis, S. Zimbeck, and M. B. Srivastava, "On communication security in wireless Ad-Hoc sensor networks," *The Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2002.

[35] L. M. Wang, J. F. Ma, and Y. B. Guo, "Node-failure tolerance of topology in wireless sensor networks," *International Journal of Network Security*, vol. 7, no. 2, pp. 261-264, 2008.

[36] A. D. Wood, and J. A. Stankovic, "Denial Of service in sensor networks," *IEEE Computer*, vol. 35, pp. 54-62, 2002.

[37] S. Wu, Y. Pawar, and N. Aeron, "Security issues in sensor network". (http://www. cs. ndsu. nodak. edu/ shwu/693-1stprogress.doc)

[38] S. Yi, P. Naldurg, and R. Kravets, *Security- Aware Ad hoc Routing for Wireless Networks,* UIUCDCS-R-2001-2241, Aug. 2001.

Garth V. Crosby received his Ph.D. and M.S. degrees in Electrical Engineering and Computer Engineering, respectively, from Florida International University. He received his B.S. in Electronics from the University of the West Indies (Mona). Currently, he is an assistant professor at Southern Illinois University Carbondale.

Dr. Crosby's research interests include wireless sensor networks, body sensor networks, network security, trust and privacy. He has served as a reviewer for several conferences, and journal publications, including IEEE INFOCOM, Wireless Communications & Mobile Computing, Ad Hoc Networks, and the International Journal of Network Security (IJNS). He is a member of IEEE Computer and Communication Societies, the National Society of Black Engineers (NSBE), and Eta Kappa Nu.

Lance Hester received his B.S. degree in Electrical Engineering from the University of Maryland College Park, and both his M.S. and Ph.D. in Electrical and Computer engineering from Northwestern University, Evanston, Illinois.

He is currently with the ACS- Wireless Group at Honey International. Prior to working at Honeywell he was at Florida Communications Research Labs of Motorola Labs for several years. His research interests include wireless communications, sensor and ad hoc wireless networks, and communication network performance evaluation.

Niki Pissinou received her Ph.D. in Computer Science from the University of Southern California, her M.S. in Computer Science from the University of California Riverside, and her B.S. in Industrial Systems and Engineering from the Ohio State University. Currently, she is a professor and the director of Florida International University (FIU) Telecommunications and Information Technology Institute (IT2).

Dr. Pissinou's research interests include distributed systems, computer networks and databases. She has served on the editorial boards of several journals including IEEE transactions on Data and Knowledge Engineering, and has been the general and program chair of a number of conferences.