*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2007 proceedings.*

# Insider Attacker Detection in Wireless Sensor Networks

Fang Liu & Xiuzhen Cheng
Department of Computer Science
The George Washington University
Washington, DC 20052
{fliu, cheng}@gwu.edu

Dechang Chen
Uniformed Services University
of the Health Sciences
Bethesda, MD 20817, USA
dchen@usuhs.mil

***Abstract*— Though destructive to network functions, insider attackers are not detectable with only the classic cryptography-based techniques. Many mission-critic sensor network applications demand an effective, light, flexible algorithm for internal adversary identification with only localized information available. The insider attacker detection scheme proposed in this paper meets all the requirements by exploring the spatial correlation existent among the networking behaviors of sensors in close proximity. Our work is exploratory in that the proposed algorithm considers multiple attributes simultaneously in node behavior evaluation, with no requirement on *a prior* knowledge about normal/malicious sensor activities. Moreover, it is application-friendly, which employs original measurements from sensors and can be employed to monitor many aspects of sensor networking behaviors. Our algorithm is purely localized, fitting well to the large-scale sensor networks. Simulation results indicate that internal adversaries can be identified with a high accuracy and a low false alarm rate when as many as 25% sensors are misbehaving.**

## I. Introduction

Security provisioning is a critical requirement for many sensor network applications (battlefield reconnaissance, homeland security monitoring, etc.). Nevertheless, the constrained capabilities of smart sensors (battery supply, CPU, memory, etc.) and the harsh deployment environment of a sensor network (infrastructureless, unattended, wireless, ad hoc, etc.) make this problem very challenging [15]. Many researchers have been working towards securing sensor networks in the fields of pairwise key establishment [13][14][16], authentication [23], access control [26], defense against attacks [29], etc.

Most of the existent works rely on the traditional cryptography and authentication techniques to establish a trustworthy relationship among the collaborative sensors. However, the unreliable wireless channels and unattended operation make it very easy to compromise/capture sensors and break the trust relationship established beforehand. Sensors are envisioned to be low-cost and lack of tamper resistance. The compromise or capture of a sensor releases all the security information to the adversary. Then, the adversary can easily launch *internal attacks* with data alteration, message negligence, selective forwarding, jamming, etc [10][20]. The *insider attackers* are severely destructive to the functioning of a network. For example, an insider attacker can easily fabricate a false event report to mislead the decision makers, or keep injecting bogus data to cause network outage, etc.

Unfortunately, internal attacks cannot be solved by the classic cryptographic techniques solely [10][11][19]. Conventional methods such as encryption, authentication, etc., have the ability to verify the correctness and integrity of an operation, but could not eliminate all attacks, especially the insider attacks. An internal adversary can easily *modify and forward* with access to the valid cryptographic keys. An insider attacker detection scheme must be designed to ensure many of the mission-critic applications.

However, detection of internal adversaries is not trivial at all. The major difficulty comes from the resource-constrained sensors and the infrastructureless network, which render it impossible to copy from the intrusion detection techniques developed for a fixed wired network. A typical low-cost sensor has limited memory budget and restricted computational capability, thus is not capable of creating and studying a detection log file to identify an internal attack. It is also impossible for a base station to collect audit data from the entire network and label malicious sensors in a centralized fashion, due to the large network size and infrastructureless architecture. An in-situ detection scheme must be designed to be localized and computationally efficient, so as to reduce bandwidth and battery consumption. Moreover, the only available resources for the detection algorithm are the communication activities occurring within a limited range, which constitutes another challenge for internal adversary identification. The algorithm design must consider how to obtain satisfactory accurate results based on the partial and localized information. Finally, the unattended operation and harsh environment make the problem even more challenging. Sensors may malfunction due to hardware crash, security attack, environment disturbance, etc. A solid malicious sensor detection algorithm must be robust and fault-tolerant.

Despite the many difficulties, detection of insider attackers may be accomplished by exploring the correlation among neighboring sensors. In a typical sensor network with collaborative in-network processing (e.g. data aggregation, etc.), sensors are expected to be burdened with similar communication and computation workloads in close proximity. On the other hand, an internal adversary usually misbehaves in some

**0743-166X/07/$25.00 ©2007 IEEE**

aspects with respect to normal sensors, such as broadcasting or dropping excessive packets, generating "abnormal readings" that deviated remarkably from a typical application-specific range, etc. Intuitively, when a significant change takes place in the networking behavior of a single sensor, this sensor should be faulty or malicious with a big chance.

Inspired from the spatial correlation existent in the neighborhood activities, we propose a localized algorithm for insider attacker detection for wireless sensor networks in this paper. Each sensor monitors the networking behaviors of immediate neighbors, with the inspection conducted regarding multiple aspects of node behaviors. In a sparse network, each sensor may also use for reference the monitoring results of neighboring nodes, with the data source selected by a trust-based node evaluation scheme. Then a neighbor is suspected to be an internal adversary if its behavior is "extreme" compared with those from the same neighborhood. The comparison is conducted by considering all the features simultaneously. The final decision is adjusted based on the detection results from the neighborhood through the majority vote. Compared to the existent works for intrusion/misbehavior detection in wireless networks [7][10][11][18], our algorithm has the following characteristics and advantages:

- Our algorithm explores the spatial correlation in neighborhood activities, and requires no prior knowledge about normal or malicious sensors. This property is important since the requirement of *a priori* knowledge not only incurs extra training overhead, but also introduces a serious concern in that attack behaviors may change dynamically and no fixed *a priori* knowledge can properly reflect this dynamism.
- Our algorithm is generic, which can monitor many aspects of sensor networking behaviors. Compared with those using a $0/1$ *decision predicate* [6][21] by comparing the measurements with a predetermined threshold, our algorithm should be more precise and more robust since the original measurements are used without any second round approximation.
- Our algorithm is localized, with the information exchange restricted in a limited neighborhood. A high detection accuracy can be obtained with a low false alarm rate, even when as many as $25\%$ misbehaving sensors are present in the network.

The paper is organized as follows. Section II summarizes the related works. We present the network model and assumptions in Section III. The localized algorithm for insider attacker detection is proposed in Section IV and analyzed in Section V. Simulation results are reported in Section VI. We conclude the paper with a discussion in Section VII.

## II. Related Works

In this section, we summarize the most related works along three major lines: detection of faulty sensor readings, detection of routing misbehavior, and detection of intrusion in wireless networks.

Detection of event region or faulty sensors is explored for $0/1$ decision predicate computation in [6][12][21]. The motivation comes from the observation that a remarkable change in sensor readings usually indicates a faulty sensor or a real event. The related algorithms require only the most recent readings (within a sliding window) of individual sensors. No collaboration among neighboring sensors are exploited. In [6], the "change point" of the time series are statistically computed. The result is used to answer questions such as "when does the front line of the contamination reach a location?" The detector proposed in [12] computes a running average and compares it with a threshold, which can be adjusted by a false alarm rate. In [21], kernel density estimators are designed to check whether the number of "abnormal" readings are beyond an application-specific threshold. The research on faulty sensor identification has been improved significantly in [9][27] by allowing any kind of scalar values as inputs instead of only $0/1$ decision predicates. The detection algorithms in [9][27] can also infer faulty sensors from event sensors and compute the boundary of the event region. Similarly, our misbehaving sensor detection algorithm accepts any inputs expressed by real numbers. Further, our algorithm advances one more step in identifying misbehaving sensors by considering multiple attributes simultaneously.

For detection of failed or routing misbehaving sensors, one solution is to leverage the route discovery and update. Common routing protocols evade failed nodes through the re-establishment of route discovery [2][22]. Base stations can also help identify routing misbehaviors [24][25]. Staddon *et al.* [24] propose to trace failed nodes in sensor networks at a base station, assuming all sensor measurements will be directed to the base station along a routing tree. In this work, the base station has a global view of the network topology, and can identify failed nodes through route update messages. In [25], base stations launch marked packets to probe sensors and rely on the responses to identify and isolate insecure locations. Our algorithm can also be employed for secure routing. However, no routing or global topology information is required, which provides better scalability and flexibility. Furthermore, our algorithm can be combined with any routing protocol to route the detected information to base stations for further instructions.

A second solution for detecting routing misbehaviors is to let sensors monitor the neighborhood activities through *watchdog*-like techniques [4][8][10][18][19]. The watchdog method is first proposed by Marti *et al.* [18], which is used to detect packet dropping attacks by letting nodes listen promiscuously to the next-hop node's broadcasting transmission. The monitoring result is used by the pathrater, which maintains a rating for the other nodes and selects a reliable route for data delivery. In [8], multiple watchdogs work collaboratively in decision making. In [4][10][19], a reputation system is constructed to provide a quality rating of participants. The monitoring result should go through the reputation system that will notify the path manager to delete the path from the path cache [4], or inform the provider to deny the execution

of the requested operation [10][19], if the rating of a node turns out to be intolerable. Though multiple misbehaviors or attacks are monitored in [4][10][19], they are not evaluated simultaneously, which is different from our work. Further, our algorithm works on the original measurement results that retain the correlation among the attributes, thus should be more precise and more robust.

Zhang *et al.* [30] are the first to work on intrusion detection in wireless ad hoc networks. A new architecture is investigated for collaborative statistical anomaly detection, which provides protection from attacks on ad-hoc routing, on wireless MAC protocols, or on wireless applications and services. In [3], in-network outlier detection is studied, where each sensor first identifies outliers in the neighborhood based on some detection function, then keeps exchanging the decisions with neighbors to obtain the global set of outliers. The update process is expensive, yet not necessary considering the fact that the collaboration among sensors is often restricted in a limited neighborhood. In [7], messages are collected in a promiscuous mode, and pre-selected rules are applied to determine if a failure happens. An intrusion alarm is raised if the number of failures exceeds a predefined threshold. In this work, multiple rules are defined, and a decision is made based on a simple summation of the rule application results. In [11], a learning-based approach is proposed for anomaly detection. Cross-feature analysis is conducted by computing classifiers from a training set composed of normal nodes. An intrusion is alarmed if the correlation between the features does not match those of the classifiers. The learning procedure assumes a large number of features being monitored from sensor behaviors, and the availability of normal sensors as the training data set, both of which are difficult to obtain considering the restrained sensor resources and dynamic networking behaviors. Our algorithm is more versatile. It works well with any number of features, and requires no prior knowledge on normal/malicious sensor activities.

## III. Network Model and Assumptions

We consider a homogeneous sensor network with $N$ sensors uniformly distributed in the network area. The network region is a $b \times b$ squared field located in the two dimensional Euclidean plane $\mathcal{R}^2$. All sensors have the same capabilities, and communicate through bidirectional links. We assume sensors in the proximity are burdened with similar workloads, thus nearby sensors are expected to behave similarly under normal conditions. An *insider attacker* is a sensor under the control of an adversary. It has the same network resource as a normal sensor, but its behavior is different compared to others. For example, an insider attacker may drop or broadcast excessive packets, report false readings that deviate significantly from other readings of neighboring sensors, etc. Throughout this paper, insider attackers are also called *outliers* or *outlying sensors*, while sensors working properly are called *normal sensors*.

We assume each sensor works in promiscuous mode intermittently and listens on the channel for activities of direct neighbors. That is to say, sensor $x$ can overhear the message to and from the immediate neighbor $x_i$ no matter whether or not $x$ is involved in the communication. The monitoring is conducted intermittently, and $x_i$'s networking behavior is modeled by a $q$-component attribute vector $f(x_i) = (f_1(x_i), f_2(x_i), ..., f_q(x_i))^T$ with each component describing $x_i$'s activity in one aspect. For each fixed $j$ $(1 \leq j \leq q)$, the component $f_j(x_i)$ represents the actual monitoring result, such as the number of packets being dropped or broadcasted in one unit time, the actual reading of temperature/light/sound, the number of occurrences of some phenomenon, and so on. Therefore, $f_j(x_i)$ can be continuous or discrete. For convenience, we assume that in any local area of the sensor field, all $f(x_i)$, where $x_i$'s are normal sensors, follow the same multivariate normal distribution. (See details on this assumption in Subsection IV-C.)

After an internal adversary is detected, a report should be generated to the base station. Each sensor should exclude the outlying sensors in selecting the next-hop forwarder to realize the secure routing. In this paper we focus on the detection of insider attackers, thus report generation/delivery to the base station and outliers isolation will not be considered. In addition, we assume there exists a MAC layer protocol to coordinate neighboring broadcastings such that no collision occurs.

## IV. Localized Insider Attacker Detection

In this section, we present our algorithm for detecting insider attackers whose behaviors are "abnormal" with respect to normal sensors. The algorithm consists of the following four phases: collecting local information, filtering the collected data, identifying initial outliers using Mahalanobis distances, and applying the majority vote to obtain a final list of outlying sensors.

### A. Information Collection

Let $\mathcal{N}_1(x)$ denote a bounded closed set of $\mathcal{R}^2$ that can be directly monitored by sensor $x$. Specifically, $\mathcal{N}_1(x)$ can be $x$'s one-hop neighborhood in watchdog-like techniques [18]. Let $\mathcal{N}(x)(\supseteq \mathcal{N}_1(x))$ denote another closed set of $\mathcal{R}^2$ that contains the sensor $x$ and additional $n-1$ nearest sensors. The set $\mathcal{N}(x)$ represents another neighborhood of $x$, whose selection is determined by the node density in the network. For a dense network, we can simply choose $\mathcal{N}(x) = \mathcal{N}_1(x)$, while for a sparse network, $\mathcal{N}(x)$ may include $x$'s two-hop neighbors. As stated in Section III, sensor $x$ should monitor the activities of sensors in $\mathcal{N}_1(x)$ and express the results using $q$-component attribute vectors. Then, the observed results are broadcasted within the neighborhood $\mathcal{N}(x)$, so that sensor $x$ obtains a set $\mathcal{F}(x)$ of attribute vectors, where $\mathcal{F}(x) = \{f(x_i) = (f_1(x_i), f_2(x_i), ..., f_q(x_i))^T | x_i \in \mathcal{N}(x)\}$.

For example, during the process of neighborhood monitoring, sensor $x$ can evaluate its immediate neighbor $x_i$ in terms of the following metrics. In this example, $q = 4$.

- *Packet dropping rate*. After sensor $x$ forwards a packet $M$ to the neighbor $x_i$, $x$ should check whether $x_i$

forwards $M$ by promiscuously listening. As implemented in watchdog [18], sensor $x$ should retain a buffer for the recently sent packets, and check if there is a match for each overheard packet. The buffer will be updated over time, and $x_i$ is considered to have dropped the packet $M$ if no match is found before $M$ is deleted from the buffer.

- *Packet sending rate*. Among all the overheard packets, sensor $x$ counts the number of packets that $x_i$ has sent out in one unit time.
- *Forwarding delay time*. This can be measured together with the packet dropping rate. $x$ measures the difference between the time $x$ sends the packet $M$ to $x_i$ and the time $x_i$ forwards $M$.
- *Sensor readings*. Each node broadcasts its sensor reading to the direct neighbors once after a specified period of time. Sensor $x$ updates its record after receiving $x_i$'s reading.

### B. False Information Filtering

After the information collection phase, sensor $x$ obtains the data set $\mathcal{F}(x)$. This set is expected to describe the true activities about the neighborhood $\mathcal{N}(x)$. However, this may not be the real case when $\mathcal{N}_1(x) \subset \mathcal{N}(x)$ and $\mathcal{F}(x)$ contains indirect monitoring results. An internal adversary may exist in $\mathcal{N}_1(x)$ and forward to $x$ a false attribute vector $f(x_i)$ about a two-hop neighbor $x_i \in \mathcal{N}(x) - \mathcal{N}_1(x)$. For an accurate detection result, such false information must be filtered as much as possible[1], which can be accomplished through the following *Trust-Based False Information Filtering Protocol*.

Based on the direct neighborhood monitoring, sensor $x$ assigns a *trust value* to each neighbor $x_i \in \mathcal{N}_1(x)$. The trust value $T(x_i)$ is in the range $[0, 1]$, where a value closer to 1 indicates a higher probability that $x_i$ is a normal sensor. In our consideration, sensors should behave similarly in the close proximity. Thus, $T(x_i)$ can be computed according to the degree of $x_i$'s deviation from the neighborhood activities.

Let $\mathcal{F}_1(x)$ denote the attribute vectors of $\mathcal{N}_1(x)$, i.e., $\mathcal{F}_1(x) = \{f(x_i) = (f_1(x_i), f_2(x_i), ..., f_q(x_i))^T | x_i \in \mathcal{N}_1(x)\}$. Let $\hat{\mu}_j, \hat{\sigma}_j$ denote the sample mean and sample standard deviation of $\mathcal{F}_1(x)$'s $j$-th component set $\mathcal{F}_{1,j}(x) = \{f_j(x_i) | x_i \in \mathcal{N}_1(x)\}$, respectively, i.e.,

$$\hat{\mu}_j = \frac{1}{n_1} \sum_{i=1}^{n_1} f_j(x_i), \quad (1)$$

$$\hat{\sigma}_j = \sqrt{\frac{1}{n_1 - 1} \sum_{i=1}^{n_1} (f_j(x_i) - \hat{\mu}_j)^2}, \quad (2)$$

where $n_1$ is the number of nodes in $\mathcal{N}_1(x)$. Sensor $x$ first standardizes each data set $\mathcal{F}_{1,j}(x)$ $(1 \le j \le q)$ and computes the absolute values to obtain $\mathcal{F}'_{1,j}(x) = \{f'_j(x_i) | x_i \in \mathcal{N}_1(x)\}$, where

$$f'_j(x_i) = |\frac{f_j(x_i) - \hat{\mu}_j}{\hat{\sigma}_j}|. \quad (3)$$

[1]There is no need to filter $\mathcal{F}(x)$ if $\mathcal{N}(x) = \mathcal{N}_1(x)$.

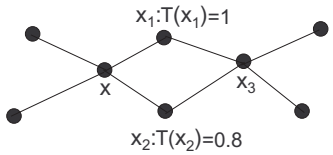



Fig. 1. After receiving two attribute vectors about $x_3$, sensor $x$ selects the one from $x_1$ since it is more "believable" based on the trust values.

For each $x_i \in \mathcal{N}_1(x)$, sensor $x$ computes the maximum attribute component $f'_M(x_i) = \max\{f'_j(x_i) | 1 \le j \le q\}$, which indicates the "extremeness" of $x_i$'s deviation from the neighborhood activities. Then, the trust value is computed as

$$T(x_i) = f^m_M / f'_M(x_i), \quad (4)$$

where $f^m_M = \min\{f'_M(x_i) | x_i \in \mathcal{N}_1(x)\}$.

As illustrated in Fig. 1, sensor $x$ may have received $t$ different attribute vectors regarding a neighbor $x_j \in \mathcal{N}(x) - \mathcal{N}_1(x)$ from $t$ direct neighbors $x_{j_1}, ..., x_{j_t} \in \mathcal{N}_1(x)$ residing between $x$ and $x_j$. A node $x_{j_T} (1 \le T \le t)$ is said to be the reliable relay node for $x_j$ if

$$T(x_{j_T}) = \max\{T(x_{j_s}) | 1 \le s \le t\}, \quad (5)$$

$$T(x_{j_T}) \ge T_{min}, \quad (6)$$

where

$$T_{min} = f^m_M / 2. \quad (7)$$

$T_{min}$ may be treated as the minimum acceptable trust value.[2] Sensor $x$ will dismiss the information about $x_j \in \mathcal{N}(x) - \mathcal{N}_1(x)$ if no reliable relay node for $x_j$ can be found in $\mathcal{N}_1(x)$. Thus after filtering $\mathcal{F}(x)$ by Eqs. (5)(6), sensor $x$ will only consider information carried by sensors from the set $\tilde{\mathcal{N}}(x)$, where $\tilde{\mathcal{N}}(x) \subseteq \mathcal{N}(x)$ and $\tilde{\mathcal{N}}(x)$ contains $x$'s direct neighbors in $\mathcal{N}_1(x)$ and $x$'s two-hop neighbors that have a trustworthy relay node in $\mathcal{N}_1(x)$. Then the new data set to be assessed by sensor $x$ is $\tilde{\mathcal{F}}(x) = \{f(x_i) = (f_1(x_i), f_2(x_i), ..., f_q(x_i))^T | x_i \in \tilde{\mathcal{N}}(x)\}$.

### C. Outlier Detection

Sensor $x$ detects if any outliers exist by studying the data set $\tilde{\mathcal{F}}(x)$. The detection is conducted by computing the distance between each sensor $x_i \in \tilde{\mathcal{N}}(x)$ to the "center" of the data set $\tilde{\mathcal{F}}(x)$. Sensor $x_i$ is determined as an outlier if the distance is larger than a predefined threshold $\theta_0$. Before going into the details of the method, we first present the following observation. First, by Section III, we assume that all $f(x_i)$ $(x_i \in \tilde{\mathcal{N}}(x))$ form a sample of a multivariate normal distribution. If $f(x_i)$ is distributed as $N_q(\mu, \Sigma)$, i.e., $q$-dimensional vector $f(x_i)$ follows a multivariate normal distribution with mean vector $\mu$ and variance-covariance matrix $\Sigma$, the Mahalanobis squared distance $(f(x_i) - \mu)^T \Sigma^{-1} (f(x_i) - \mu)$ is distributed as $\chi^2_q$, where $\chi^2_q$ is the chi-square distribution

[2]Assuming the behavior of a component can be modeled by the standard normal distribution $N(0, 1)$, then with a probability of $95.45\%$, the behavior of the component should fall into the range $[-2, 2]$.

with $q$ degrees of freedom. Therefore the probability that $f(x_i)$ satisfies $(f(x_i)-\mu)^T\Sigma^{-1}(f(x_i)-\mu) > \chi^2_q(\alpha)$ is $\alpha$, where $\chi^2_q(\alpha)$ is the upper $(100\alpha)$-th percentile of a chi-square distribution with $q$ degrees of freedom. Hence, for a sensor $x_i$, if $(f(x_i)-\mu)^T\Sigma^{-1}(f(x_i)-\mu)$ is sufficiently large, $x_i$ should be treated as an insider attacker.

Now for the data set $\tilde{\mathcal{F}}(x) = \{f(x_i)|x_i \in \tilde{\mathcal{N}}(x)\}$, sensor $x$ estimates the location $\mu$ and variance-covariance matrix $\Sigma$. Let $\hat{\mu}$ and $\hat{\Sigma}$ be the estimate of $\mu$ and $\Sigma$, respectively. Then the probability of $f(x_i)$ satisfying $(f(x_i)-\hat{\mu})^T\hat{\Sigma}^{-1}(f(x_i)-\hat{\mu}) > \chi^2_q(\alpha)$ is expected to be roughly $\alpha$. Let $d(x_i) = (f(x_i)-\hat{\mu})^T\hat{\Sigma}^{-1}(f(x_i)-\hat{\mu})^{1/2}$. Sensor $x_i$ will be treated as an outlier if $d(x_i)$ or $d^2(x_i)$ is unusually large. In our scheme, sensor $x$ declares $x_i$ as an outlier if $d^2(x_i) > \theta_0$. Bellow we discuss estimation of $\mu$ and $\Sigma$, as well as determination of $\theta_0$.

*1) Computation of $\hat{\mu}$ and $\hat{\Sigma}$:* Clearly, a simple solution is to estimate $\mu$ and $\Sigma$ by the sample mean and sample variance-covariance matrix of $\tilde{\mathcal{F}}(x)$, i.e.,

$$\hat{\mu} = \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} f(x_i), \tag{8}$$

$$\hat{\Sigma} = \frac{1}{\tilde{n}-1}\sum_{i=1}^{\tilde{n}}[f(x_i)-\hat{\mu}][f(x_i)-\hat{\mu}]^T, \tag{9}$$

where $\tilde{n}$ is the number of nodes in $\tilde{\mathcal{N}}(x)$. However, it is well known that the sample mean and sample variance-covariance matrix in Eq. (8)(9) may not be reliable, since they are sensitive to the presence of outliers. The values from outlying sensors can easily distort the estimates of $\mu$ and $\Sigma$, and the detection via Mahalanobis distances will fail to identify true outlying sensors. Therefore, robust estimators $\hat{\mu}$ and $\hat{\Sigma}$ are required, which are expected to be less influenced by outliers and thus generate estimates close to the true values of $\mu$ and $\Sigma$. Throughout this paper, we employ the *Orthogonalized Gnanadesikan-Kettenring* (OGK) estimators $\hat{\mu}$ and $\hat{\Sigma}$ [17], as described below.

We begin with the univariate case. Let $Y = \{y_1, y_2, ...., y_n\}$ be a single-variate sample set coming from a distribution with mean $\mu$ and variance $\sigma^2$. Let $\mu_0$ and $\sigma_0$ be the median and MAD[3] of $Y$, respectively. Define a weight function $W(x) = (1-(x/c_1)^2)^2 I(|x| \le c_1)$ and a $\rho$-function $\rho(x) = min(x^2, c_2^2)$, where $c_1 = 4.5$ and $c_2 = 3$. Then $\mu$, $\sigma^2$ can be estimated by [28]:

$$\hat{\mu} = \frac{\sum_{i=1}^n y_i W(v_i)}{\sum_{i=1}^n W(v_i)} \text{ for } v_i = \frac{y_i-\mu_0}{\sigma_0}, \tag{10}$$

$$\hat{\sigma}^2 = \frac{\sigma_0^2}{n}\sum_{i=1}^n \rho(\frac{y_i-\hat{\mu}}{\sigma_0}), \tag{11}$$

respectively.

Now we describe the OGK estimates $\hat{\mu}$ and $\hat{\Sigma}$ based on the multivariate data set $\tilde{\mathcal{F}}(x) = \{f(x_i) = (f_1(x_i), f_2(x_i), ..., f_q(x_i))^T | x_i \in \tilde{\mathcal{N}}(x)\}$. Let $\hat{\mu}(\cdot)$ and $\hat{\sigma}(\cdot)$ denote the univariate statistics, as described in Eq (10)(11). The OGK estimates can be computed as follows:

1) Compute $\mathcal{G}(x) = \{g(x_i)|x_i \in \tilde{\mathcal{N}}(x)\}$ from $\tilde{\mathcal{F}}(x)$, where $g(x_i) = P^{-1}f(x_i)$ for $P = diag(\hat{\sigma}(\tilde{\mathcal{F}}_1(x)), \hat{\sigma}(\tilde{\mathcal{F}}_2(x)), ..., \hat{\sigma}(\tilde{\mathcal{F}}_q(x)))$. Here $\tilde{\mathcal{F}}_j(x)$ is the $j$-th component set of $\tilde{\mathcal{F}}(x)$, with $\tilde{\mathcal{F}}_j(x) = \{f_j(x_i)|x_i \in \tilde{\mathcal{N}}(x)\}$, $1 \le j \le q$.
2) Calculate a $q \times q$ matrix $R$, with $R_{j,k}$, the element at the $j$th-row and $k$-th column defined as
$$R_{j,k} = \begin{cases} \frac{1}{4}[\hat{\sigma}^2(\mathcal{G}_j+\mathcal{G}_k) - \hat{\sigma}^2(\mathcal{G}_j-\mathcal{G}_k)] & \text{if } j \ne k, \\ 1 & \text{if } j = k. \end{cases} \tag{12}$$
3) Apply the spectral decomposition to obtain $R = Q\Lambda Q^T$, where $Q$ is the $q \times q$ matrix whose columns are the eigenvectors of $R$, and $\Lambda$ is the diagonal matrix composed of $R$'s eigenvalues.
4) Compute $\mathcal{H}(x) = \{h(x_i)|x_i \in \tilde{\mathcal{N}}(x)\}$ from $\mathcal{G}(x)$, where $h(x_i) = Q^T g(x_i)$. Then calculate $\Delta = (\hat{\mu}(\mathcal{H}_1(x)), \hat{\mu}(\mathcal{H}_2(x)), ..., \hat{\mu}(\mathcal{H}_q(x)))^T$, and $\Gamma = diag(\hat{\sigma}^2(\mathcal{H}_1(x)), \hat{\sigma}^2(\mathcal{H}_2(x)), ..., \hat{\sigma}^2(\mathcal{H}_q(x)))$. Here $\mathcal{H}_j(x)$ denotes the $j$-th component set of $\mathcal{H}(x)$.
5) Let $V = PQ$. The robust estimates of multivariate location and dispersion are $\hat{\mu} = V\Delta$ and $\hat{\Sigma} = V\Gamma V^T$, respectively.

Different solutions have been proposed in the literature to generate robust estimates, which are used to help calculate reliable Mahalanobis distances in the presence of outliers. However, the application of most of these estimates is restricted either by a low breakdown point[4] (e.g. M-estimates), or by high computational overheads (e.g. MCD, MVD, SDE, P-estimates etc.) [17]. We choose OGK since it ensures a high breakdown point at the expense of a much lower computational cost. OGK computes the multivariate dispersion estimates based on pairwise robust correlation or covariance estimation, which reduces the computation complexity in the data dimension $q$ from exponential ($2^q$) to quadratic ($q^2$) [1].

*2) Determination of the threshold $\theta_0$:* After calculating the Mahalanobis distance for each neighbor in $\tilde{\mathcal{N}}(x)$, sensor $x$ should produce a list of suspicious neighbors, which is denoted by $\mathcal{D}(x)$. For this purpose, one option is to simply select $k$ nodes with the largest $k$ Mahalanobis distances from $\tilde{\mathcal{N}}(x)$, where $k = \tilde{n} \times p_o$ with $p_o$ denoting the sensor outlying probability. The estimation of $p_o$ can be studied using empirical data, which is often difficult to obtain. Another selection method focuses on checking whether or not $d^2(x_i) > \theta_0$. We note that robust estimates introduced above are less influenced by the values of outlying sensors, and can provide more accurate assessment of the mean and covariance of the population based on the normally working sensors. For this reason, we will use $\chi^2_q(\alpha)$, the percentile of the chi-square distribution with $q$ degrees of freedom as the threshold $\theta_0$. Thus, $x_i$ will be selected as an outlier if and only if $d^2(x_i) > \chi^2_q(\alpha)$. This approach will be adopted in our simulation studies.

[3] $MAD(Y) = median(|Y - median(Y)|)$.

[4] A breakdown point is defined to be the maximal proportion of outliers that the estimator can tolerate.

| $\tilde{n}$ | $ID_1$ | $Status_1$ | ... ... | $ID_{\tilde{n}}$ | $Status_{\tilde{n}}$ |
|---|---|---|---|---|---|

$\tilde{n}$ : *the number of sensors in* $\tilde{N}(x)$

$ID_i$: *the ID of the sensor* $x_i$

$Status_i$: *x's decision about whether or not* $x_i$ *is an outlier (1/0)*

Fig. 2. The message format of the outlier announcement from sensor $x$.

### D. Majority Vote

Now sensor $x$ obtains a set of suspicious nodes $\mathcal{D}(x)$ from Phase IV-C. These sets can be combined through the majority vote to reach the final decision regarding whether or not a sensor is outlying.

To begin, each sensor $x$ will announce all identified outlying neighbors $\mathcal{D}(x)$ to a neighborhood $\mathcal{N}^*(x)$, where $\mathcal{N}(x) \subseteq \mathcal{N}^*(x)$. Using a larger set $\mathcal{N}^*(x)$ ensures that more neighbors will participate in voting. As illustrated in Fig. 2, the broadcasting message includes $x$'s evaluation on a neighbor $x_i \in \tilde{\mathcal{N}}(x)$, with $x_i$'s status as $1/0$ indicating that $x_i$ is $outlying/normal$. At the same time, sensor $x$ will receive announcements from others and should make records of all the votes regarding to its neighbors in $\mathcal{N}(x)$. Then for a neighbor $x_i$, sensor $x$ counts the proportion $p_i$ among all the received advertisements that $x_i$ is an outlier and decides finally that $x_i$ is an insider attacker if $p_i > 0.5$. Such a decision can be combined with the routing protocol to select a reliable next-hop forwarder. Further, a report can be generated and delivered to the base station, which should isolate the insider attacker $x_i$ if multiple reports about $x_i$ have been received.

In general the majority vote combines decisions from various resources to reach a result with a higher accuracy. Below we present one analysis on the performance of the majority vote. Consider the sensor $x$, and let $p$ denote the posterior probability of being abnormal for the given sensor $x$. For simplicity, we assume that $p > 0.5$, i.e., $x$ is an outlier. (The case $p \leq 0.5$ can be discussed similarly.) Suppose that $l$ sensors $s_1, s_2, \cdots, s_l$ are available to assess the status (outlying/normal) of $x$ using the procedure presented in Subsection IV-C. It is not hard to see that the decision from $s_i$ can be formulated as a fraction $y_i \in (0, 1)$ that is used to estimate $p$. Therefore these $l$ sensors contribute the following estimates of $p$: $y_1, y_2, \cdots, y_l$. Let $y_1, y_2, \cdots, y_l$ be rearranged in order from least to greatest and let the ordered values be $y_{(1)}, y_{(2)}, \cdots, y_{(l)}$, where $y_{(1)} \leq y_{(2)} \leq \cdots \leq y_{(l)}$. Suppose the sequence $y_1, y_2, \cdots, y_l$, denoted by $[y]$, forms a sample from a distribution function. Then the effect of majority vote applied to $y_1, y_2, \cdots, y_l$ is equivalent to that of the following function of the sample:

$$V([y]) = \begin{cases} y_{((n+1)/2)} & \text{if } n \text{ is odd} \\ y_{(n/2)} & \text{if } n \text{ is even.} \end{cases} \quad (13)$$

The decision regarding the status of $x$ is made by $V([y])$ in the following way: $x$ is an outlier if and only if $V([y]) > 0.5$. Note that $V[y]$ is a random variable whose distribution affects the decisions. It can be shown that, under certain conditions [5], $V([y])$ is asymptotically normally distributed, and the probability of error in classifying $x$ by the majority vote is

$$\lim_{l\to\infty} e(V[y]) = \lim_{l\to\infty} P(V([y]) \leq 0.5) = 0.$$

This shows that the results from voting will be more accurate when more sensors participate in voting.

The majority vote is essential for our detection scheme in that it not only enlarges a sensor's field of vision and makes the final decision more accurate, but also helps prolong the network lifetime. Note that the underlying MAC protocols usually implements a sleep-wakeup schedule to save energy, which makes continuous monitoring almost impossible. With the majority vote, a sensor can use its neighbor's monitoring results for reference. Thus, a sensor listens intermittently for the neighborhood activities and makes its decision by integrating the observations from the neighborhood. The majority vote makes our detection scheme applicable in the resource-restrained sensor networks.

## V. Performance Analysis

### A. Computation Complexity

The computation overhead comes mainly from the last three phases. Assume a $q$-component vector is available to model a sensor's behavior. Let $n, n_1$ and $m$ denote the number of sensors in the neighborhood $\mathcal{N}, \mathcal{N}_1$ and $\mathcal{N}^*$, respectively. In false information filtering, the attribute standardization costs $O(qn_1)$. Finding the maximum component attributes $\{f'_M(x_i)|1 \leq i \leq n_1\}$ costs $O(qn_1)$, while finding the minimum $f_M^m$ and computing the trust values both cost $O(n_1)$. To select a reliable intermediary node for each node in $\mathcal{N}-\mathcal{N}_1$, the computation cost is $O(nn_1)$. Hence, the computation cost of the second phase is $O(nn_1)$ if $q \ll n$. In the third phase, sensor $x$ first computes the OGK estimates $\hat{\mu}$ and $\hat{\Sigma}$ at the cost of $O(q^2n)$, then calculates the Mahalanobis distances for each neighbor in $\tilde{\mathcal{N}}(x)$ at the cost of $O(q^3n)$. Thus, the computational complexity for the third phase is $O(q^3n)$. In the last phase, the majority vote costs $O(mn)$. Thereafter, the computation complexity of the detection algorithm is $O(mn)$ if $m \gg n \gg q$.

### B. Communication Overhead

Our detection algorithm involves only localized data exchange. In the information collection phase, sensor $x$ broadcasts its monitoring results within the neighborhood $\mathcal{N}(x)$. In the voting phase, an outlier advertisement is broadcasted within the neighborhood $\mathcal{N}^*(x)$. Thus the communication overhead is $O(n) + O(m)$, which is approximately $O(m)$ if $m \gg n$.

Note that the selection of $\mathcal{N}(x)$ and $\mathcal{N}^*(x)$ is dependent on the network density. With a dense network, $\mathcal{N}(x)(\mathcal{N}^*(x))$ can be chosen as $x$'s 1-hop (2-hop) neighborhood, which incurs the least communication overhead. For a sparse network with a low density, $\mathcal{N}(x)$ and $\mathcal{N}^*(x)$ should contain the multi-hop neighborhood for better performance. In this case, a higher communication overhead is generated to obtain good detection

results. This represents a tradeoff between communication overhead and performance.

## VI. Simulation Study

### A. Simulation Settings

We consider a $64 \times 64$ square region in our simulation study. $N = 4096$ sensors are uniformly distributed in the network region. The behavior of each sensor is modeled by a vector containing $q = 3$ attributes. For normal sensors, the attribute values are drawn from $N_3(\mu_1, \Sigma_1)$; for outlier sensors, the attribute values are drawn from $N_3(\mu_2, \Sigma_2)$. In our simulations, we set $\mu_1 = (\mu_{11}, \mu_{12}, \mu_{13}) = (10, 15, 20)$, $\mu_2 = (\mu_{21}, \mu_{22}, \mu_{23}) = (30, 35, 40)$, $\Sigma_1 = \Sigma_2 = \Sigma$, where the variance-covariance matrix $\Sigma$ is defined using the standard deviations $\sigma_i$ and the correlation coefficient matrix $\rho$[5]. We select $(\sigma_1, \sigma_2, \sigma_3) = (1, 1, 1)$ and consider the following three cases for $\rho$:

$$\rho_1 = \begin{bmatrix} 1 & 0.9 & 0.9 \\ 0.9 & 1 & 0.9 \\ 0.9 & 0.9 & 1 \end{bmatrix}, \quad (14)$$

$$\rho_2 = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 1 & -0.5 \\ -0.5 & -0.5 & 1 \end{bmatrix}, \quad (15)$$

$$\rho_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (16)$$

where $\rho_1$ indicates large positive correlation, $\rho_2$ indicates medium negative correlation, $\rho_3$ indicates independence among attributes. This presents a coarse summary of all the possibilities in the real network scenarios. Note that the means and variances can be selected arbitrarily as long as the difference $|\mu_{1j} - \mu_{2j}|$ $(1 \le j \le q)$ is large enough compared with the related standard deviations.

We assume that outlying sensors are distributed uniformly amongst normal sensors. An outlying sensor may be "abnormal" in one single aspect of its networking behavior, or in all aspects. We also assume that when broadcasting an observation about the neighbor $x_i$, as stated in Section IV-A, an outlying sensor $x$ may modify each attribute value $f_j(x_i)$ $(1 \le j \le q)$ with a probability $p_e$, which is set to be $p_e = 0.5$ in our simulations. The modification is simulated by adding a noise $e$ as

$$f_j(x_i) \leftarrow f_j(x_i) + e, \quad (17)$$

where $e$ is drawn from $N(0, \sigma_e^2)$ with $\sigma_e^2 = 100$. We then observe that the above selection of values of $\mu_1, \mu_2$ and $\sigma_e^2$ makes it possible that a normal attribute is modified to an outlying one, or vice versa. This might indicate one aspect of the behavior of a "smart" adversary.

We consider the application of our algorithm in both sparse and dense networks, and conduct two tests correspondingly. For both cases, $\mathcal{N}_1(x)$ is chosen to be $x$'s one-hop neighborhood. $\mathcal{N}(x)$ is selected to be $x$'s two-hop neighborhood for a sparse network in Test 1, and $\mathcal{N}(x) = \mathcal{N}_1(x)$ for a dense network in Test 2. The results are averaged over 100 runs for both tests.

In all the simulations, we evaluate our detection algorithm in terms of the following metrics:

- *Detection accuracy*: the ratio of the number of insider attackers detected to the total number of insider attackers.
- *False alarm*: the ratio of the number of normal sensors that are claimed as insider attackers to the total number of normal sensors.

Both metrics are in the range $[0, 1]$. The higher the detection accuracy and the lower the false alarm, the better the detection algorithm.

### B. Simulation Results

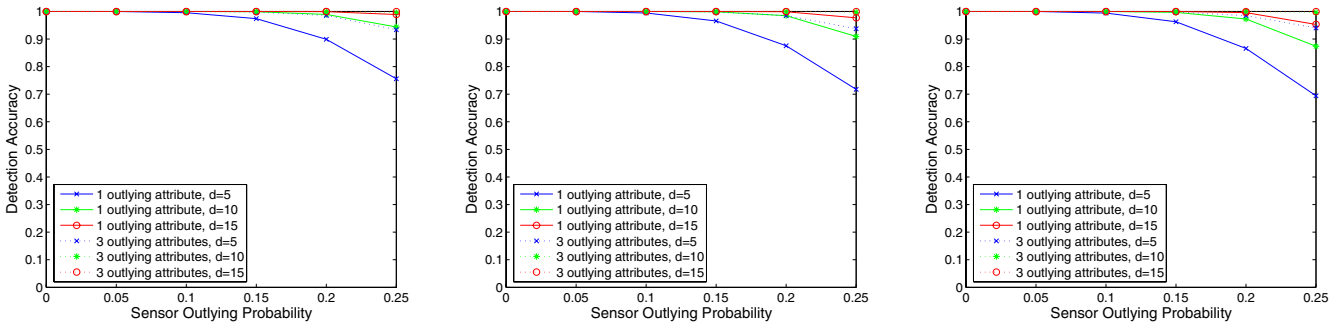
As illustrated in Figs. 3 and 4, our algorithm can effectively identify insider attackers with a high detection accuracy and a low false alarm in most cases. Let $d$ denote the average number of direct neighbors, which also represents the node density in the network. Our detection algorithm can reach a high detection accuracy ($> 90\%$) when as many as $25\%$ sensors are outlying, for a sparse network with $d \ge 10$, and for a dense network with $d \ge 25$. We observe that a higher $d$ value results in a better detection accuracy. The improvement comes from the increase in the size of the sample data set, since a larger size implies that there is more information used to estimate the location and dispersion of the distribution (Subsection IV-C) and more information available to the voting operation (Subsection IV-D).

Another important feature of our detection scheme is the robustness. We observe that the increase in the number of outlying sensors leads to no increase in the false alarm rate. Such a nice property results from the robust statistics employed in the computation of Mahalanobis distances. Note that with more malicious sensors, more bogus information will be injected, and consequently the functioning of the detection algorithm is more likely to be disturbed. However, as the number of outlying sensors increases, our algorithm is robust in that its performance degrades very slowly in the detection accuracy and it restrains the false alarm rate effectively.

In general, the detection accuracy decreases along with the increase of the sensor outlying probability. Such a trend is much more obvious when a sensor misbehaves in only one aspect of its networking behavior. However, when an outlying sensor is abnormal in all aspects, the detection accuracy can be as high as 1 in most cases. This improvement is due to the more "obvious" deviation, and our detection algorithm is "sensitive" to the degree of an outlier's deviation from the normal nodes.

Another interesting observation from Figs. 3 and 4 is that the correlation among the attributes influences the detection accuracy, especially when a misbehaving sensor has only one outlying attribute. Though the differences are not significant, we can still observe that the more correlated the attributes, the

[5]Given a $q \times q$ correlation coefficient matrix $\rho = (\rho_{ij})$ and a $1 \times q$ vector of standard deviations $\sigma_i$, the variance-covariance matrix $\Sigma = (\Sigma_{ij})$ can be determined by $\Sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$, $1 \le i, j \le q$.




(a) Detection Accuracy: $\rho = \rho_1$ (b) Detection Accuracy: $\rho = \rho_2$ (c) Detection Accuracy: $\rho = \rho_3$

(d) False Alarm Rate: $\rho = \rho_1$ (e) False Alarm Rate: $\rho = \rho_2$ (f) False Alarm Rate: $\rho = \rho_3$

Fig. 3. Test 1: Sparse networks.

better the detection accuracy. This pattern may help design a special detection scheme for some specific attributes. We will explore along this direction in our future study.

## VII. Conclusion and Discussion

In this paper we propose a novel idea of insider attacker detection in wireless sensor networks. By exploiting the spatial correlation among the networking behaviors of sensors in close proximity, our detection algorithm can achieve a high detection accuracy and a low false alarm rate as indicated by the extensive simulation study. The nice feature of the algorithm is that it requires no prior knowledge about normal or malicious sensors, which is important considering the dynamic attacking behaviors. Further, our algorithm can be employed to inspect any aspects of networking activities, with the multiple attributes evaluated simultaneously. The algorithm is pure localized, thus scales well to large sensor networks.

We notice that the detection algorithm can be specialized by exploring the degree of the correlations existent among different aspects of sensor networking behaviors. We target this specialization as a future work.

## Acknowledgment

The research of Dr. Xiuzhen Cheng is supported by the NSF CAREER Award No. CNS-0347674.

D. Chen was supported by the National Science Foundation grant CCR-0311252.

## References

[1] F. A. Alqallaf, K. P. Konis, R. Douglas Martin, Ruben H. Zamar, "Scalable robust covariance and correlation estimates for data mining", ACM SIGKDD 2002, pp.14-23, Edmonton, Alberta, Canada.

[2] V. Bhuse, A. Gupta, L. Lilien, "Detection of packet dropping attack for wireless sensor networks," in *the 4th International Trusted Internet Workshop (TIW)*, India, December 18-21, 2005.

[3] J. W. Branch, B. K. Szymanski, C. Giannella, R. Wolff, H. Kargupta, "In-Network Outlier Detection in Wireless Sensor Networks," in *IEEE ICDCS'06*, Lisboa, Portugal, July 2006.

[4] S. Buchegger, J. Le Boudec, "Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes - Fairness in Dynamic Ad-hoc Networks," in *ACM MOBIHOC 2002*, Lausanne, Switzerland, June 2002.

[5] D. Chen and X. Cheng, "An asymptotic analysis of some expert fusion methods," *Pattern Recognition Letters*, 22, pp. 901-904, 2001.

[6] D. Chen, X. Cheng, and M. Ding, "Localized Event Detection in Sensor Networks," *manuscript*, 2004.

[7] A. P. da Silva, M. H. Martins, B. P. Rocha, A. A. Loureiro, L. B. Ruiz, H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," in *ACM Q2SWinet'05*, 2005.

[8] A. Deshpande, A. Hegde, A. Shetty, "CVS: Collaborative Voting System to detect routing misbehavior in wireless ad hoc networks," in *ICENCO04*, Cairo, Egypt, December 27-30, 2004.

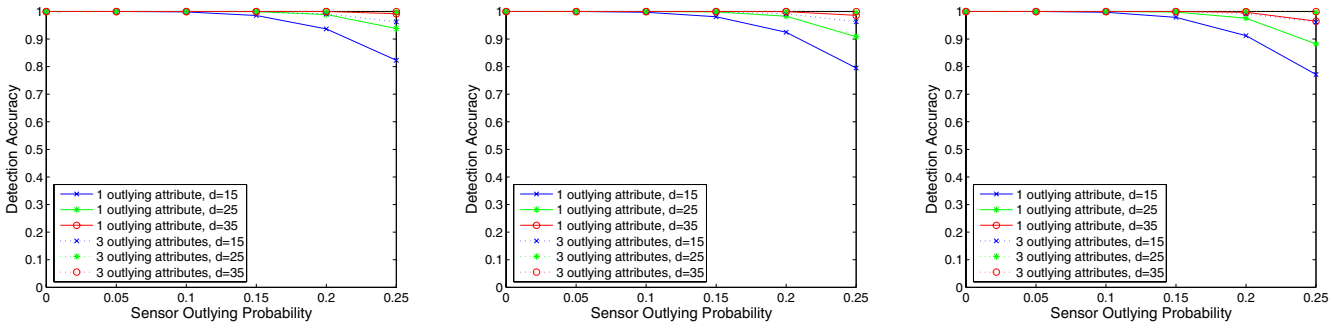
[9] M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized Fault-Tolerant Event Boundary Detection in Sensor Networks," in *IEEE INFOCOM 2005*, March 13-17, 2005.

[10] S. Ganeriwal, M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," in *ACM SASN'04*, Washington, DC, October 25, 2004.

[11] Y. Huang, W. Lee, "A Cooperative Intrusion Detection System for Ad Hoc Networks," in *ACM SASN'03*, Fairfax, VA, October 2003.

[12] D. Li, K.D. Wong, Y.H. Hu, and A.M. Sayeed, "Detection, Classification, and Tracking of Targets," *IEEE Signal Processing Magazine*, Vol. 19, pp. 17-29, March 2002

[13] F. Liu, X. Cheng, "LKE: A Self-configuring Scheme for Location-aware Key Establishment in Wireless Sensor Networks," to appear in *IEEE Transactions on Wireless Communications*, 2006.




(a) Detection Accuracy: $\rho = \rho_1$ (b) Detection Accuracy: $\rho = \rho_2$ (c) Detection Accuracy: $\rho = \rho_3$

(d) False Alarm Rate: $\rho = \rho_1$ (e) False Alarm Rate: $\rho = \rho_2$ (f) False Alarm Rate: $\rho = \rho_3$

Fig. 4. Test 2: Dense networks.


[14] F. Liu, X. Cheng, "A Self-Configured Key Establishment Scheme for Large-Scale Sensor Networks," in *IEEE MASS 2006*, Vancouver, Canada, October 9-12, 2006.

[15] F. Liu, X. Cheng, F. An, "On the Performance of In-Situ Key Establishment Schemes for Wireless Sensor Networks," in *IEEE GLOBECOM 2006*, San Francisco, CA, November 27-Decrmber 1, 2006.

[16] L. Ma, X. Cheng, F. Liu, J. Rivera, F. An, "iPAK: An In-Situ Pairwise Key Bootstrapping Scheme for Wireless Sensor Networks," to appear in *IEEE Transactions on Parallel and Distributed Systems*, 2006.

[17] R. A. Maronna, R. D. Martin, V. J. Yohai, *Robust Statistics: Theory and Methods*, Wiley Publisher, 2006.

[18] S. Marti, T.J. Giuli, K. Lai, M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *ACM MOBICOM 2000*, pp. 255-265, Boston, USA, August 2000.

[19] P. Michiardi, R. Molva, "CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Proc. IFIP 6th Joint Working Conference on Communications and Multimedia Security (CMS02)*, pp. 107-122, Portoro, Slovenia, September 2002.

[20] P. Ning, K. Sun, "How to Misuse AODV: A Case Study of Insider Attacks against Mobile Ad-hoc Routing Protocols," in *Ad Hoc Networks*, Vol. 3, No. 6, pp. 795-819, November 2005.

[21] T. Palpanas, D. Papadopoulos, V. Kalogeraki, D. Gunopulos, "Distributed Deviation Detection in Sensor Networks," in *Sigmod Record* 32(4), Special Issue on Sensor Technology, December 2003.

[22] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, Vol. 8, No. 5, pp.521-534, 2002.

[23] K. Ren, W. Lou, Y. Zhang, "LEDS: providing location-aware end-to-end data security in wireless sensor networks," in *IEEE INFOCOM 2006*, Barcelona, Spain, April 2006.

[24] J. Staddon, D. Balfanz, and G. Durfee, "Efficient tracing of failed nodes in sensor networks," in *WSNA 2002*, pp. 122-130, Atlanta, USA.

[25] S. Tanachaiwiwat, P. Dave, R. Bhindwale, A. Helmy, "Location-centric Isolation of Misbehavior and Trust Routing in Energy-constrained Sensor Networks," in *EWCN 2004*, Phoenix, Arizona, April 14-17, 2004.

[26] H. Wang, B. Sheng, Q. Li, "Elliptic Curve Cryptography Based Access Control in Sensor Networks," in *International Journal of Sensor Networks*, 2006.

[27] W. Wu, X. Cheng, M. Ding, K. Xing, F. Liu, P. Deng, "Localized Outlying and Boundary Data Detection in Sensor Networks," to appear in *IEEE Transactions on Knowledge and Data Engineering*, 2006.

[28] V. J. Yohai, R. Zamar, "High Breakdown-Point Estimates of Regression by Means of the Minimization of an Efficient Scale", *Journal of the American Statistical Association*, Vol.86, No.402, pp.403-413, June 1988.

[29] Y. Zhang, W. Liu, W. Lou, Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE Journal on Selected Areas in Communications, Special Issue on Security in Wireless Ad Hoc Networks*, Vol. 24, No. 2, pp. 247-260, February 2006.

[30] Y. Zhang, W. Lee, "Intrusion Detection in Wireless Ad-hoc Networks," *ACM MOBICOM 2000*, pp. 275-283, Boston, USA, August 2000.