Trust based Approach for Improving Data Reliability in Industrial Sensor Networks

Tatyana Ryutov and Clifford Neuman
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001, Marina del Rey, CA 90292

Abstract. The resource constraints and unattended operation of wireless sensor networks make it difficult to protect nodes against capture and compromise. While cryptographic techniques provide some protection, they do not address the complementary problem of resilience to corrupted sensor data generated by failed or compromised sensors. Trusting data from unattended sensor nodes in critical applications can have disastrous consequences. We propose a behavior-based trust mechanism to address this problem in static sensor networks, in which the location of nodes is known. We take advantage of domain knowledge which includes: (i) physical constraints imposed by the local environment where sensors are located, (ii) expectations of the monitored physical phenomena; and (iii) sensor design and deployment characteristics. The system diagnoses and isolates faulty/malicious nodes even when readings of neighboring nodes are faulty. The goal of this system is to increase work effort and capabilities required by an attacker. The framework and related techniques of behavior-based trust are discussed in this paper.

1 Introduction

Sensor network technology has great value for many industrial applications, including oil and gas production, industrial plant monitoring and maintenance [1]. Use of permanent sensors mounted on industrial equipment enables facilities to gather operational data and to send it to analytical tools that examine critical operating parameters (e.g., casing gas pressure, temperature, pump torque, etc). This enables proactive management of operations by adjusting settings to maintain steady state conditions.

While remote asset monitoring and control dramatically enhance operating efficiencies, use of untrusted data from unattended sensor nodes in critical applications can have disastrous consequences. An inherent assumption that all nodes are trusted leaves the nodes at the mercy of an adversary who can insert faulty data by exploiting access in the physical environment (e.g., placing a source of heat close to a sensor) or by compromising the sensor. In the absence of adequate physical and cyber security, an adversary can mislead a process control system responsible for procedures that are critical to productivity and safety of the plant facilities.

While cryptographic techniques [3], [4], [7], [13] and [20] make sensor networks more secure, they do not address the complementary problem of resilience to corrupted sensor data generated by failed or compromised sensors. The difficult issue that needs to be addressed is falsification of sensor data due to node capture, compromise or abuse of the physical environment.

Because sensed events are ambiguous with respect to causes, diagnosing normal and malicious/faulty sensor behavior in a distributed industrial environment is a difficult technical problem. Consider, for example, a Weatherford's optical flow meter system that provides real-time measurements of oil flow rate in a section of a production pipe. The meter employs an array of sensors mounted on the outside of the flow pipe to measure the velocity and speed of sound of the flowing stream. The differences in the incoming and outgoing flows may suggest that (i) some of the sensed data is corrupted or (ii) there is an oil leak. Since the pipeline might be situated in a physically unprotected area, an attacker can easily compromise the external sensors in order to report a false oil leak event or, even worse, hide a real one.

Current approaches [5], [6], [8], [9], [10], [11], [14] and [21] for detecting and correcting malicious/faulty sensor readings suffer from reliance on the node neighbors. A common underlying assumption about sensor faults being uncorrelated is not practical. An attacker could compromise a number of sensors located in a physically insecure place, or some natural event could impact a group of sensors in close proximity to the event. Accurate real time detection of malicious/faulty sensors requires contextual information, such as deployment parameters, baseline system and sensor behavior, and underlying process models.

This paper targets the identification of malicious/faulty sensors and detecting abnormal events in static, context aware sensor networks deployed in industrial facilities. In such environments, the location of a single node is known and the spatial temporal correlations in the underlying physical process are known as well. We propose a behavior-based trust solution that ensures that only trusted sensor readings are accepted even when a group of neighboring sensors misbehaves. The goal of this system is to increase the work effort and capabilities required by an attacker. With this system in place, an attacker must simultaneously compromise a number of sensors of different types deployed at various locations. Some of the locations can be physically protected, be difficult to reach or unknown to the attacker. The effects of the compromise are contained to be those that effectively duplicate the expected relationships between the readings of different sensor groups. However, the behavior of maliciously cooperating nodes will be different from the arbitrary nature of failure of faulty but non-malicious nodes. With this knowledge, we can find correlations between compromised sensors belonging to different groups and detect malicious node collaborations.

The main contributions of this paper can be summarized as follows: (1) a scheme for representing and evaluating trust (suspicion levels) based on conformance of empirical observations to a set of expectations; (2) an approach for defining and representing contextual information (expectations); (3) a methodology for determining data trustworthiness and updating suspicion levels even when the readings of neighboring nodes are faulty.

2 The Trust Model

In this section, we describe our approach to representing and computing trust given a set of pre-defined expectations and direct observations of sensor performance.

2.1 User Expectations and Trust

In static industrial environments, sensor nodes are deployed to monitor a particular facility. We can build a set of accurate expectations using extensive domain knowledge: deployment parameters, baseline facility and sensor behavior, and underlying process models. This is different from more dynamic applications of sensor networks that observe unknown environments (e.g., habitat monitoring). In such cases we may not have exact knowledge of the phenomenon behavior or pre-collected sensor readings.

We need mathematical tools to represent expectations, continuously confirm system performance to the expectations based on direct observations and finally, make a transition from confirmation level to trust metric of a node. Trust is an overloaded term used with a variety of meanings in different contexts. In our approach, we are concerned with a particular type of trust - **trust in behavior** that reflects strict conformance to a set of pre-defined **expectations**:

- Sensor nodes report correct real world readings that reflect the behavior of the observed facility;
- Sensor nodes confirm appropriate behavior consistent with the sensor design characteristics (e.g., expected sending rate, sensing radius, natural error rate);
- Readings of sensors monitoring different aspects of physical phenomena (e.g., pressure/temperature, torque/flux) must conform to temporal and spatial dependencies according to the expectations that we have developed based on past experiences and laws of physics.

2.2 Suspicion Level as a Metric of Distrust

We associate a Suspicion Level (SL) with each sensor. SL represents the belief that the sensor is not acting reliably according to the expectations of sensor behavior formed before the actual interactions with the sensor. During the interaction, the system assesses perceived sensor performance against the original expectations and determines the extent to which the expectations are confirmed. When results are not as expected, the system increases the SL for the sensor of concern.

SL for a node is a variable taking values on the interval (0, 1], that represents the accumulative performance of the node in past interactions. To calculate a SL for a node S_i during the evaluation phase *N,* we adopt the approach described in [8]. Assume that the natural error rate for the node S_i is $0<NER<<1$. The system maintains a nonnegative variable α for each node that is used to update the SL during each evaluation phase *N*. Each time a node does not act according to the expectations, its α is incremented by $1\text{-}NER^{Si}$. Each time a node behavior is assumed correct, its α is decreased by NER^{Si}. Thus correctly functioning nodes will have a SL approaching 0 while faulty and malicious nodes will have a higher SL. The SL is calculated as:

$$SL_N^{S_i} = 1 - e^{-\omega_{S_i}\alpha_N}, \alpha_N = \alpha_{N-1} + (1 - NER^{S_i}) \;\; or \;\; \alpha_N = \max(0, \alpha_{N-1} \; - NER^{S_i}) \quad \textbf{(1)}$$

Here ω^{Si} is a proportionality constant that depends on the sensor S_i design and deployment parameters. Sensor data may have different “value” to an end user depending on the sensor design and deployment characteristics, such as reliability and the data paths used to obtain data from the sensor. These characteristics are represented by ω^{Si} for each sensor and are used to bootstrap initial SL values for each sensor. At system initialization time $N=0$, the SL assigned to each node S_i is given by an initialization function *f*: $SL_0^{S_i} = f(\omega^{S_i})$.

Note that ω^{Si} influences the convergence of SL to distrust: the higher the value ω^{Si}, the more suspicious we are when we detect that node S_i misbehaves. Figure 1 shows the effects of different values taken by ω^{Si} on the shape of the function $SL_N^{S_i} = 1 - e^{-\omega_{S_i}\alpha_N}$. Krasniewski et al [8] show that SL for an uncompromised node is expected to remain at the same value.

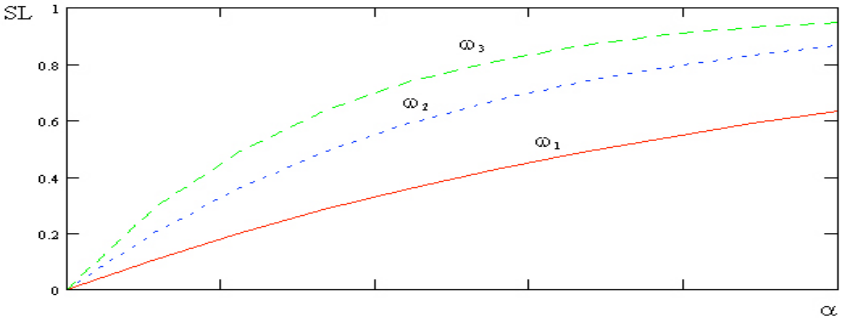

Fig. 1. The effects of ω ($\omega_1<\omega_2<\omega_3$) on the shape of the function $SL_N^{S_i} = 1 - e^{-\omega_{S_i}\alpha_N}$

In this paper, we consider the following types of malicious sensor behavior that causes the system to increase the SL of a sensor:

- False event reports (e.g., false fire alarm);
- Not reported events (e.g., an oil leak);
- Excessive reports;
- Incorrect reports: incorrect data (e.g., temperature, pressure) or event.

Note that if a node faults once, it does not mean that the node is considered faulty/malicious for the rest of the time. If the subsequent readings of the node are assumed valid by the system, the suspicion level will decrease according to the formula (1).

SL is the core of our behavior-based trust framework. Process control and security policies that govern operation of a monitored facility are conditioned on suspicion levels in order to deemphasize results from malicious or faulty sensors. The readings of a sensor with a high value of SL are treated with suspicion. If node's SL exceeds a certain threshold (specified in the policies), the node is deemed untrustworthy and is blacklisted.

3 Basic System Design

We consider a sensor network that consists of individual sensors permanently mounted on industrial equipment. The nodes of different types sense real world phenomena (e.g., temperature, pressure, flux) and forward observed measurements to a base station. We place trust with the base station, which has sufficient resources to resist attacks and is located in a physically secure place. The individual sensors are not trusted. The underlying network assumptions include the following:

- sensors are stationary (immobile);
- the location of each node is known;
- sensors can be securely and uniquely identified;
- each message is integrity protected and authenticated.

To make these assumptions practical, we leverage prior efforts. In particular, SPINS [13] protocols can provide secure sensor authentication and message integrity protection. The location of each sensor can be securely verified using the mechanisms described in [12].

3.1 Categories of Expectations

We use deployment knowledge, baseline facility behavior, and sensor design parameters to construct internal representations of the three categories of expectations:

1. **Expected Individual Sensor Behavior** is represented as follows according to sensor design characteristics and deployment parameters:

$$E^{S_i}=\{NER, RR, DR, L, r\},$$

where *NER* - natural error rate that nodes are allowed to make due to natural causes, $0<NER<<1$;

RR - reporting rate, i.e. expected number of reports per specified time interval;

DR - data range of values that the reported data can take;

L - position of the sensor;

r - sensing radius within which a node can detect the occurrence of an event or take measure. We refer to the sensing area of a node *S* as a sphere with a sensing radius *r* centered at the location of the node.

2. **Expected Sensor Group Behavior** describes dependencies between data reported by different groups of sensors.

Sensor data redundancy

We assume that a phenomenon of a particular type can be correctly sensed by *n* sensors of the same type which form a group of neighbors G_i due to redundant node placement. The system maintains information about all groups and the membership in the groups. This is possible due to the static nature of the nodes.

Consider a simple example of a sensor network deployed in a pipeline system used to transport crude oil to a refinery (Figure 2). The network is tasked with pipeline monitoring and oil leak detection. The pump pushes the crude oil into the pipeline. The pipeline monitoring system ensures leak detection by either observing presence of oil in the surrounding area or by measuring pressure and flux simultaneously. When oil is transmitted in an encapsulated pipe, the flux at both ends of the oil-transporting pipes should remain steady [19].

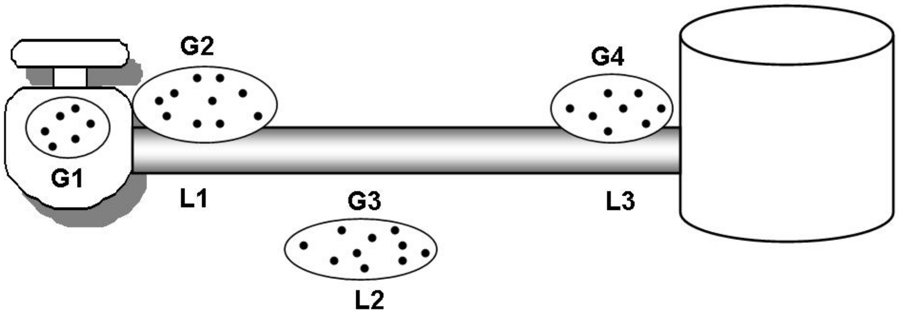

Fig. 2. The Pipeline Example

In our example, there are four sensor groups installed in the field equipment:

1) G_1 measures speed of the pump's impeller, it reports data D^{G1};
2) G_2 measures oil flux in the pipe at the location L_1, it reports data D^{G2};
3) G_3 detects the presence of oil in the soil at the location L_2, it reports data D^{G3};
4) G_4 measures oil flux in the pipe at the location L_3, it reports data D^{G4}.

Reading of sensors comprising group G_3 are binary: $D^{G3} = 1$ if G_3 detects oil in the soil at the location L_2, otherwise $D^{G3} = 0$. The readings of all other sensor groups are continuous. We expect that all sensors within a group must report the same data, but are allowed to make errors only within a specified bound defined by *NER*.

Complimentary Analysis

We consider temporal and spatial dependencies between different groups of sensors in order to detect anomalies due to faulty or malicious nodes. In our example, measurements reported by groups G_1 and G_2 provide complementary analyses and verifications of the pipeline operation. If the pump speed sensor indicates an unusually high speed, the flux sensors at the location L_1 must display a corresponding anomaly. Alternatively, if the torque sensor indicates a "normal" level of torque, the flux must be within a normal range. Deviations of data reported by one of the groups indicate a potential problem with the sensors.

The expectations about temporal and spatial correlations are encoded as a set of relationships. A relationship $R(e_1, e_2) \rightarrow T$ *or* F is a Boolean function that indicates dependency between two entities e_1 and e_2. An entity can represent readings of a group or another relationship. A *simple relationship* links readings of two groups of sensors. A *compound relationship* either links readings of a group of sensors with another relationship, or relates two other relationships. A relationship R holds if it evaluates to T (a Boolean true), R does not hold if it evaluates to F (a Boolean false).

Some relationships must always hold due to, for example, laws of physics or empirical knowledge. We call such relations - *ground relationships*. The fact that a ground relationship does not hold means that one of the groups included in the relation is lying.

In our pipeline example, we define three relationships:

1) $R_1(D^{G1}, D^{G2})$ is a temporal, simple, ground relationship that relates the pump speed and flux measurements reported by groups G_1 and G_2. The pump must obey the pump law: the oil flux is directly proportional to the pump speed, therefore the changes in speed and flux must be proportional [19].

2) $R_2(D^{G2}, D^{G4})$ is a temporal, simple relationship that relates the levels of oil flux measured by the groups of sensors G_2 and G_4 at the locations L_1 and L_3. Under normal pipeline operation, the flux at both locations must be equal.

3) $R_3(D^{G3}, R_2)$ is a temporal, ground, compound relationship that describes the dependencies between the results of soil tests for oil contamination at the location L_2 and the differences in flux at the locations L_1 and L_3. R_3 holds if the levels of flux at L_1 and L_2 are equal within an acceptable error range.
   $R_3 \rightarrow T$ if:
   1) $D^{G3} = 0$ and $R_2 \rightarrow T$, **OR**
   2) $D^{G3} = 1$ and $R_2 \rightarrow F$.

   Otherwise, $R_3 \rightarrow F$.

3 **Expected Facility Behavior** describes the expected system behavior based on prior experience and desired facility state (e.g., baseline flux/pressure/torque range).

For each group of sensors, the expected facility behavior describes a range of expected values:

D^{G1}should be within normal pump speed range $[S_1, S_n]$;
$D^{G2}$and D^{G4} should be within normal flux range $[F_1, F_n]$;
$D^{G3} = 0$ (no oil contamination should be reported).

The system detects abnormal state of the monitored facility by comparing the observed readings (perceived trustworthy) against the Expected Facility Behavior. If any anomalies are detected, the system reacts according to policies that govern the operation of the monitored facility (e.g., adjust pump speed).

3.2 Determining Data Trustworthiness and Updating SLs

The system evaluates sensor data and updates SLs for all sensors periodically. Each evaluation phase consists of two steps. First, the system considers each individual sensor and updates corresponding SL if the sensor behavior does not comply with the expectations. Next, the system averages the data reported by each group and calculates SL for each group. The system employs the averaged data to determine which ground relationships do not hold and which groups report wrong data. For the groups deemed to be wrong, the SLs of each sensor in the group are further increased. During the third step the system acts according to trust-based policies. We next describe each step in detail.

Step 1

During each evaluation phase *N*, the system waits for a predefined interval of time Δ to receive reports from deployed nodes. After Δ has elapsed, the system assesses the conformance of each sensor to the Expected Individual Sensor Behavior E^{S_i}. Some types of malicious/faulty behavior can be detected at this step. For example, if the report sending rate *RR* considerably exceeds the expected rate, the system may suspect a Denial of Service Attack (DoS) on behalf of the sensor and take an appropriate action. On the other hand, if perceived *RR* is lower than expected, the system may suspect an aging node due to battery energy depletion. If the sensor readings are out of the expected data range *DR,* the data is assumed invalid and is not used for evaluation. In all these cases, the SL for the node is updated according to the formula (1). Note that a node may continue to report correct data even if we suspect a DoS or a dying battery. If we know that the node might have been compromised or is aging, we will treat the data from that node with suspicion, until it can be manually verified that the node is benign.

However, not all types of abnormal behavior (discussed in Section 2.2) can be detected at this point. For example, detecting false alarms or incorrect reports (e.g., false temperature reading) requires additional knowledge. So, the next step is to employ the Expected Sensor Group Behavior in order to determine the quality of the data reported by each node.

Step 2

Binary Data

For each group G_i which reports binary events, the system decides whether an event actually occurred by partitioning the sensors into two sets based on whether the sensor reported the occurrence of the event or not. The SLs of sensors in each set are summed and the set with the lower cumulative SL wins [8]. If the group G_i reading is decided to be correct during this step, the suspicion levels of the sensors comprising the faulty set is increased according to formula (1).

Continuous data

For each group G_i consisting of n sensors, during each evaluation stage N the system calculates the weighted mean of the data $D_N^{G_i}$ reported by $m \le n$ sensors. m is less or equal to n because (i) some sensors might have missed their reports; (ii) data reported by certain sensors fell out of the expected data range DR and was discarded, as described above.

$$D_N^{G_i} = \frac{\sum_{j=1}^{m} SL_N^{S_j} D_N^{S_j}}{\sum_{j=1}^{m} SL_N^{S_j}} \quad \textbf{(2)}$$

For each node S_j in the group G_i the system uses a threshold δ to determine whether the node reading is correct. If the absolute difference between the sensor reading and weighted mean exceeds the threshold $| D_N^{G_i} - D_N^{S_j} | > \delta$, the data is assumed incorrect. If the system decides that the group G_i reading is accurate during this step, SLs for the incorrect nodes are updated according to the formula (1).

Next, the system employs a decision tree approach to determine whether the expectations about group and facility behavior hold. The system assumes that during each data evaluation phase N, only one sensor group can lie. Under this assumption, the decision tree either detects the lying group (if any) or provides us with a list of candidates. The decision tree is constructed a priori (e.g., using automatic decision tree construction tools such as ID3 which employs greedy top-down algorithm [15]). An internal node denotes a test on a relationship. A branch represents an outcome of the test: binary True or False. Leaf nodes represent outcome class labels, such as facility is in normal state, oil leak is detected, particular sensor group is lying.

The expectations about group and facility behavior are used to construct a decision tree. The most common method for learning decision trees is top-down induction: start from the entire set of training examples (relationships and corresponding outcome classes), partition it into subsets by testing whether a relationship holds, and then recursively call the induction algorithm for each subset [15]. The constructed tree is stored along with a set of functions that implement methods for determining whether relationships hold.

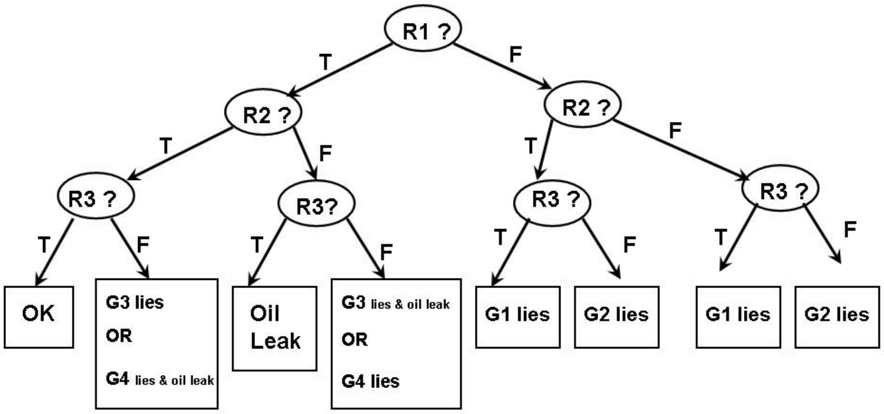

Fig. 3. The Decision Tree for the Pipeline Example

Figure 3 shows a decision tree constructed for our pipeline example. Note that R_1 and R_3 are ground relationships and must always hold; R_2 may or may not hold. If R_2 does not hold and neither group G_2, nor group G_4 is lying, then there is an oil leak. There are two situations when we can not tell whether group G_3 or G_4 is lying. Consider two cases:

1) $R_1 \rightarrow T$, $R_2 \rightarrow T$ and $R_3 \rightarrow F$
2) $R_1 \rightarrow T$, $R_2 \rightarrow F$ and $R_3 \rightarrow F$

Case 1: if G_3 lies (it reports 1 while R_2 holds), it means that the facility is in the normal state and G_3 reported a false oil leak. If we assume that G_4 lies, then R_2 does not hold and there is an oil leak.

Case 2: if G_3 lies (it reports 0 while R_2 does not hold), it means that the there is an oil leak (G_3 must have had reported 1). If we assume that G_4 lies, then R_2 and R_3 hold and the facility is in the normal state.

To decide which group from the candidate set is faulty the system calculates a SL for each group by taking a mean of the SLs of all nodes in the group calculated during the step 1.

$$SL_N^{G_i} = \frac{1}{n}\sum_{i=1}^{n} SL_N^{S_i} \quad \textbf{(3)}$$

Next the system compares the SL for each candidate group and decides that the one with the highest SL is lying.

Breaking the ties

If SLs differ insignificantly, we utilize a simple sequential parameter estimation method as a tie breaker. Note that this method works only for groups who report continuous data. The method compares the observed and predicted relationship between sensor group measurements and determines the expected group value. We assume that behavior of continuous sensor readings follows a normal distribution. This approach works if the observed phenomena are spatially and temporally consistent, so that the measurements at neighboring sensors within the same group report common information. This assumption is reasonable for industrial environments where sensors monitor environmental variables such as air temperature, humidity, flux, and soil moisture.

The normal density function [2] is:

$$p(D_N^{G_k}) = \frac{1}{(2\pi\sigma^2)}\exp\left\{-\frac{(D_N^{G_k}-\mu)^2}{2\sigma^2}\right\} \quad \textbf{(4)}$$

From the maximum likelihood estimate we can find the mean and variance of a normal distribution in one dimension:

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} D_N^{G_k\ n}, \hat{\sigma}^2 = \frac{1}{N}\sum_{n=1}^{N}(D_N^{G_k\ n} - \hat{\mu})$$ [2], or alternatively:

$$\hat{\mu}_{N+1} = \hat{\mu}_N + \frac{1}{N+1}(D_N^{G_k\ N+1} - \hat{\mu}_N)$$

$$\hat{\sigma}^2_{N+1} = \hat{\sigma}^2_N + \frac{1}{N+1}(D_N^{G_k\ N+1} - \hat{\mu}_{N+1})^2$$

For each group G_k from the candidate set, we calculate the probability $p(D_N^{G_k})$of observing data $D_N^{G_k}$ according to formula (4). The group with the lowest probability is decided to be faulty. Note that we do not have to store the complete set of sensor data since each data point can be discarded once it has been used and only $\hat{\mu}$ and $\hat{\sigma}^2$are saved. In our pipeline example, G_3 reports binary data and G_4 reports continuous data. If the values of the suspicion levels $SL_N^{G3}$and $SL_N^{G4}$are very close, we calculate $p(D_N^{G_4})$. If the probability p is lower than a threshold γ, we decide that G_4 is faulty.

Note that this method of breaking the ties must be used with caution. To confuse the system, an adversary needs to compromise most of the sensors in one of the groups (which is a valid assumption in our paper) and make the suspicion levels of the two groups equal. In our example, if an adversary compromises the majority of the sensors in the group G_3, the system cannot tell G_3 or G_4 is lying, as analyzed in Figure 3. Second, when resorting to maximum likelihood estimate, the probability of data reported by G_3 in (4) can still be high. Therefore, if the suspicion levels of the groups are low, the system should report this case to an operator. Ideally, sensor group types, placements, and the corresponding set of relationships should be defined to eliminate (or at least reduce) the number of inconclusive cases.

Next, the system updates SLs according to formula (1). For all groups deemed to be faulty, the system increases SL for each sensor in the group. For all groups who reported correct data, the SL of each sensor from the correct set is decreased.

Step 3

The system employs trust-based security policies [15], [16] to activate fine-grained real time responses (conditioned on SLs). For example, the system may blacklist sensors that are believed to send corrupted data or report incorrect/false events. However, if the system suspects a Denial of Service attack on behalf of a sensor, just blacklisting the sensor does not solve the problem because the links will still be saturated with malicious traffic. To address this issue, the reaction should isolate the misbehaving node and drop messages originated by this node.

Mistrusted nodes can regain trust dynamically over the time by reporting data that the system assumes valid. As the result, the SL will gradually decrease. However, the treatment of blacklisted nodes must be carefully regulated by domain and organization specific policies. The system should report blacklisted nodes to an operator for analysis. Most of the problems that cause node banishment require a manual resolution. For example, a non-malicious node was blacklisted because its battery was dying and needed replacement, or a tilted light sensor needed correct orientation. The blacklisted compromised sensors require physical adjustment or replacement. For some blacklisted nodes policies can specify a predetermined time period after which the nodes are unlocked with a high SL value assigned. Another issue that policies should consider is the number of blacklisted nodes in a particular group. If the number is large the policy can require a manual action. If just a few nodes were isolated, the policy can employ the time out mechanism discussed above.

4 Related Work

Considerable attention has been given to developing localized, distributed methods for fault recognition is sensor networks. The majority of these approaches rely on the neighboring nodes to calculate reputation, trustworthiness, opinion and other classes of trust-related metrics.

Krasniewski et al [8] designed a TibFit protocol to diagnose and mask arbitrary node failures in event-driven wireless sensor networks. The protocol determines whether a binary event has occurred and the location of the event by analyzing reports from the event neighbors. TibFit maintains a Trust Index for each node that represents the reliability of previous event reports of that node. Our notion of suspicion level extends the concept of the trust index. We maintain a SL for sensors which report binary and continuous data. Furthermore, the SL is updated based on the conformance of node behavior to individual- and group-level expectations.

Elnahraway et al [5] present an approach to handling outliers and missing information in sensor networks based on exploiting contextual information of the networks. This information includes spatial dependencies between spatially adjacent nodes as well as the temporal dependencies between history readings of the same sensor node. The context is used by each sensor to locally predict its current readings given its own past readings and current readings of the neighbors. This work is the closest to our approach because it relies on contextual information to detect fault readings. However, in our approach the context includes relationships among different groups of sensors (not just two neighboring sensors). We also maintain a distrust metric that allows us to deemphasize data reported by untrusted nodes.

Ganeriwal et al [6] developed a reputation system for sensor networks that uses a Bayesian formulation for reputation representation, updates, integration and trust evolution. Each node monitors behavior of other nodes and builds their reputation over time in order to characterize them as cooperative or non-cooperative. The problem of what constitutes co-operative or non-cooperative behavior has not been sufficiently explored. In our paper we explicitly specify non-compliant node behavior as non-conformance to the set of expectations.

Krishnamachari and Iyengar [9] proposed a solution to the recognition of faulty sensor readings based on a combination of shortest-path routing, and the construction of a spanning tree as a clustering mechanism. This work assumes that a node can rely on its neighbors to accept its own reading as correct if at least half of its neighbors have the same reading. Larkey et al [11] present a distributed algorithm for detecting measurement errors and inferring missing readings in sensor networks based on statistical distributions of differences between sensor readings and the readings of its neighbors.

Distributed fault-tolerance for event detection using the assumption of spatial correlation is considered in [10]. The sensor measurements are assumed to be spatially correlated. Using this principle, faulty readings are eliminated. For fault recognition, the assumption is made that sensor faults are uncorrelated. This assumption is unrealistic. It is possible that all the sensors in a particular area fail due to some external event, and generate faulty readings.

Trappe et al [18] present a high-level framework for assessing the trustworthiness of the data reported by sensors. A monitor applies consistency checks to sensed data to determine the reliability of the data. The processed data is tagged with a class (suspicious or reliable) and confidence (how sure the monitor is) values. The consistency checks may examine relationships between several physical properties. The framework is discussed at a very high level, lacking a language for expressing consistency rules and rules for updating the confidence level. In our work, we explicitly define the relationship between different sensor readings. Furthermore, we assess the reliability of the data based on the trustworthiness of the sensor. Dynamic assessment and update of a SL for each sensor allows us to detect and rule out misbehaving sensors. Such stateful SL calculation increases system resilience to compromised/faulty nodes.

Pirzada and McDonald [14] introduce a trust model that evaluates the reliability of routes in ad-hoc networks, using only direct node observations. Trust is calculated by analyzing different categories of the events, such as received, forwarded and overheard packets. The categories signify the specific aspect of trust that is relevant to a particular relationship.

Zourdaki et al [21] propose a conceptual framework for trust establishment with respect to reliable packet delivery in the presence of potentially malicious nodes. They introduce a concept of trustworthiness which combines the computed trust metric and statistical confidence associated with a trust value. Trustworthiness is computed using a Bayesian method based on observations of packet forwarding behavior by neighbor nodes.

5 Conclusions and Future Work

We presented a system that diagnoses and isolates faulty/malicious nodes even when the readings of neighboring nodes are faulty. We map the problem of identifying and isolating faulty/compromised sensors to the problem of comparing observed behavior to a set of expectations, making inferences, and updating the suspicion level associated with each sensor. The suspicion level is used to deemphasize results collected from untrusted sensors. Observed deviations from expected behavior help us to detect sensor errors or system malfunctions.

Our future work includes experiments with the system in a simulated environment and extending the framework. In our current approach, we do not take into account differences in the reporting rates of different sensors. In wireless sensor networks, it is critical to conserve energy by minimizing the idle listening time with asymmetric sleep or activity duty cycles. This approach may lead to variations of active/sleep duty cycles of different types of sensors: some sensors report almost constantly, others only according to a schedule or after an explicit query. These incompatible time scales will be taken into account when assigning a static SL and weighting the adjustment of dynamic SL to reduce the bias.

Failures due to compromised nodes can be correlated. This is different from the arbitrary nature of failure of faulty nodes. We plan to develop an approach that will look for correlations between mistrusted sensors to detect malicious node collaboration. Currently, we assume that only one group could be faulty if a ground relationship does not hold. We will consider a more complex situation where more than one group could be wrong and will develop a set of rules and constraints to determine the faulty groups.

6 Acknowledgements

This research was supported by funding the National Science Foundation under grants no. CCR-0325951 and ACI-0325409. The views and conclusions contained

herein are those of the authors and should not be interpreted as representing the official policies or endorsement of the funding agencies.

7 References

1. R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, L. Krishnamurthy, N. Kushalnagar, L. Nachman, M. Yarvis. Design and Deployment of Industrial Sensor Networks: Experiences from the North Sea and a Semiconductor Plant, Sensys 2005.

2. C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, ISBN: 0198538642, 1995.

3. M. Bohge and W. Trappe. An Authentication Framework for Hierarchical Ad Hoc Sensor Networks. In Proc. of ACM workshop on Wireless Security, 2003.

4. W. Du, J. Deng, Y. S. Han, P. Varshney, J. Katz, A. Khalili, A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. The ACM Transactions on Information and System Security, 2005.

5. E. Elnahrawy, and B. Nath. Context-aware sensors. In Lecture Notes in Computer Science, H. Karl, A. Willig, and A. Wolisz, Eds. Vol. 2920. Springer-Verlag, 77–93, 2004.

6. S. Ganeriwal and M. B. Srivastava, Reputation-based framework for high integrity sensor networks. In Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks, 66-77, 2004.

7. G. Gaubatz, J.-P. Kaps, and B. Sunar. Public key cryptography in sensor networks revisited, 1st European Workshop on Security in Ad-Hoc and Sensor Networks, LNCS 3313, 2004.

8. M. Krasniewski, P. Varadharajan, B. Rabeler, S. Bagchi, Y. C. Hu. TibFit: Trust Index Based Fault Tolerance for Arbitrary Data Faults in Sensor Networks. In Proceedings of the International Conference on Dependable Systems and Networks, 2005.

9. B. Krishnamachari and S. Iyengar, Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. IEEE Transactions on Computers, Vol.53, No.3, 2004.

10. B. Krishnamachari, S. Iyengar. Efficient and Fault-Tolerant Feature Extraction in Wireless Sensor Networks, Proceedings of Information Processing in Sensor Networks, 2003.

11. L. B. Larkey and A. A. Hagberg and L. M. A. Bettencourt. In-Situ Data Quality Assurance for Environmental Applications of Wireless Sensor Networks, Report LA-UR-06-1117, 2006.

12. D. Liu, Peng Ning and W. Du. Attack-Resistant Location Estimation in Sensor Networks. In Proceedings of The Fourth International Conference on Information Processing in Sensor Networks, 2005.

13. A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks, Wireless Networks Journal , 8(5):521–534, 2002.

14. Pirzada and C. McDonald. Establishing trust in pure ad-hoc networks, In Proceedings of the 27th Australasian Computer Science Conference, 47-54, 2004.

15. J. R. Quinlan. Induction of decision trees. Machine Learning 1:81–106, 1986.

16. T. Ryutov, L. Zhou, N. Foukia, C. Neuman, T. Leithead, K. E. Seamons. Adaptive Trust Negotiation and Access Control for Grids. In proceedings of the Grid 6th IEEE/ACM International Workshop on Grid Computing, 2005.

17. T. Ryutov, L. Zhou, C. Neuman, T. Leithead, and K. Seamons. Adaptive Trust Negotiation and Access Control. In Proceedings of SACMAT, 2005.

18. W. Trappe, Y Zhang, and B. Nath. MIAMI: Methods and Infrastructure for the Assurance of Measurement Information, In DMSN, 2005.

19. S. Q. Zhang, S J Jin, F. L Yang, X Q Wang and Q. Y. Bai. Crucial Technologies of Oil Transporting Pipe Leak Detection and Location Based on Wavelet and Chaos, 7th ISMTII2005(UK), 2005.

20. S. Zhu, S. Setia, S. Jajodia, and P. Ning. An Interleaved Hop-by-Hop Authentication Scheme for Filtering of Injected False Data in Sensor Networks, Proceedings of IEEE Symposium on Security and Privacy, 2004.

21. C. Zouridaki, B. L. Mark, M. Hejmo, R. K. Thomas. A quantitative trust establishment framework for reliable data packet delivery in MANETs, SASN 2005.