RDAS: Reputation-based Resilient Data Aggregation in Sensor Network

Carlos R. Perez-Toro, Rajesh K. Panta, Saurabh Bagchi
Dependable Computing Systems Lab (DCSL)
School of Electrical and Computer Engineering, Purdue University
{perezcr,rpanta,sbagchi}@purdue.edu

***Abstract*—Data aggregation in wireless sensor networks is vulnerable to security attacks and natural failures. A few nodes can drastically alter the result of the aggregation by reporting erroneous data. In this paper we present RDAS, a robust data aggregation protocol that uses a reputation-based approach to identify and isolate malicious nodes in a sensor network. RDAS is based on a hierarchical clustering arrangement of nodes, where a cluster head analyzes data from the cluster nodes to determine the location of an event. It uses the redundancy of multiple nodes sensing an event to determine what data should have been reported by each node. Nodes form part of a distributed reputation system, where they share information about other node's performance in reporting accurate data and use the reputation ratings to suppress the reports from malicious nodes. RDAS is able to perform accurate data aggregation in the presence of individually malicious and colluding nodes, as well as nodes that try to compromise the integrity of the reputation system by lying about other nodes' behavior. We show that RDAS is more resilient to security attacks with respect to accuracy of event localization than the baseline data aggregation protocol with no security feature.**

I. Introduction

As Wireless Sensor Networks (WSNs) start to be used for critical applications, security of the network becomes a prominent concern. The traditional cryptographic mechanisms are not sufficient for securing WSNs. Sensor nodes are deployed for long periods of autonomous operation, often in unmonitored places. This makes it possible for an adversary to physically take control of a node and all the cryptographic keys available on the node. Therefore the node can appear as a legitimate node from a crypto standpoint while performing malicious actions. The security community for WSNs has therefore developed a suite of mechanisms to complement cryptographic techniques chiefly through behavior-based detection. Such detection must fit within energy, bandwidth, memory, and computational constraints imposed by the devices. Several researchers have recently proposed the use of reputation systems as a method for behavior-based identification of malicious nodes in WSNs [1]–[4]. A *reputation system* in the WSN context is a system where the actions of every node are observed and evaluated by the other nodes in an attempt to determine their trustworthiness. The interaction a node has with another node is guided by the state (the reputation) that the node has built up about the interacting partner node. The reputation systems for WSNs are all distributed in nature since it is unrealistic to assume the existence of a single privileged node that is able to observe every other node and is the trustworthy repository of a global reputation information.

The traditional architecture of a WSN is to employ a data aggregator which performs aggregation operations on individual data collected by the sensors embedded in the sensing field. The aggregation operations could include summarization over time, computing averages across different nodes, changing the resolution of the data, etc. Data aggregation is widely favored due to the significant energy savings that result from it compared to a flat network in which each sensing node communicates with the base station. However, data aggregation is widely regarded as being sensitive to attacks and failures [5]. If a node, a set of nodes, or worse still, a set of colluding nodes, lies about the sensed data, it can significantly shift the outcome of data fusion, triggering false events, hiding true events or reducing the accuracy of the location of the sensed event. Since the base station receives an aggregation instead of the raw data it loses the ability to filter out erroneous reports. Further, the aggregator is vulnerable to security attacks as well and becomes a single point of failure.

In this paper we present the design of a protocol called **R**eputation-based Resilient **D**ata **A**ggregation **S**ystem (RDAS) that develops a distributed reputation system and applies it for secure data aggregation in the face of unreliable and malicious nodes. There exist several challenges in the effort. First, a malicious node that participates in the reputation system can degrade the system's fidelity by lying. A compromised node can falsely accuse well-behaving nodes or falsely praise misbehaving nodes. To maintain its integrity the reputation system must be able to filter out such reports. Second, the reputation system has to discriminate between legitimate nodes having occasional natural errors and malicious actions. This is impossible to do on an individual action but needs to be achieved in aggregate. Third, the distributed views of all the nodes must be reconciled to achieve joint goals of the network, such as, determining the location of a sensed event.

To meet the challenges, we extend existing reputation systems in two fundamental ways. First, we consider different functionalities of the different nodes (like sensing, aggregation) in establishing their reputation values. Second, we develop a mechanism to use the distributed reputation ratings being maintained at the different nodes. In RDAS, this takes the form of periodically electing reputable *cluster heads* (CHs) as aggregators, electing shadow CHs to observe potential misbehavior of the CH, and using the reputation ratings of the cluster nodes at the CHs for the aggregation operation. While our work is similar in goals and technique to a prior work [6], it eliminates an important assumption in the prior work, namely that all nodes have an identical view of a sensed event. We discuss further points of difference in the Related Work section. Furthermore, previous works focus either on analytic formulation of reputation based framework without considering implementation constraints (e.g., [4], [7]), or system implementation which does not address all three challenges mentioned earlier in this section (e.g., [1]–[3]). Analytic works use legitimate and anomalous behavior of the sensor nodes to build the reputation framework. But most of them are silent on what specific actions should the nodes use to infer knowledge about anomalous and legitimate behavior. Practical system works, on the other hand, focus on few components of the reputation system and generally use scaled-down version of the analytic work to fit the solution within computation, memory, and bandwidth constraints of the sensor nodes. For example, [3] assumes only binary values are being voted on and the aggregator is trusted. RDAS presents a complete functional reputation-based system, adapting and expanding existing mathematical formulations to achieve a commonly-needed functionality in WSN (data aggregation).

We use event localization as the example aggregation operation. We implement RDAS in TinyOS for Berkeley mote class of devices and run the implementation using the TOSSIM simulator. The use of an insecure data aggregation that does not use reputation (LEACH) serves as the baseline. The main contributions of this paper are the following: 1) RDAS develops and effectively uses a reputation system to add resiliency to the common function of event localization. 2) It provides concrete answers to the questions of how to effectively generate, propagate and use reputation ratings such that RDAS can handle both colluding and non-colluding faulty nodes as well as lying nodes trying to compromise the reputation system. 3) RDAS can provide security in a network where there does not exist any trusted node and under compute, memory and bandwidth constraints.

II. Related Work

Much research has been devoted to securing data aggregation in clustering-based wireless sensor networks. The work closest to ours is [6]. Like RDAS, [6] also designs a reputation scheme for data aggregation in sensor networks. However, our work differs from this work in three substantive ways. Unlike RDAS, [6] assumes that all nodes have the same view of the event — be they the sensor nodes, the aggregators, or the cluster heads. This means their readings are the same, discounting for statistical fluctuations, which they capture using a normal distribution. This assumption is central to their scheme, e.g., it is used by the sensor nodes and the cluster heads to build reputation about the aggregators. In realistic settings, there will be redundancy in sensed events with multiple nodes being capable of sensing a given event. However, it is unlikely that they will have the same readings. For many real-world sensors that have a long range, the readings can actually vary significantly. The key insight that we use is that even in the face of such variation, the variation for legitimate nodes is *predictable*, based on the amplitude of the event and locations of the sensors. Also, in [6], the authors assume that a node will behave identically in performing its data gathering function and in reporting on other nodes' behaviors. We do not make this assumption, as a nod to sophisticated adversaries, and therefore use the distinct notions of reputation and trust. Finally, in RDAS, compromised cluster heads can be detected and removed under more general circumstances than in [6]. Specifically, [6] relies on the above assumption of identical views at all the nodes, while in RDAS, cluster monitors with differing views from that of the cluster head can isolate a compromised cluster head.

In [5], the authors present an approach which combines random sampling with interactive proofs to approximate true sensor values when the aggregator and a fraction of the sensor nodes are compromised. Their approach focuses mainly on combating compromised aggregators and assumes only a small fraction of sensor nodes are corrupted. The authors of [2] present a general framework for maintaining reputation in a distributed fashion. In [1], the authors tackle the problem of false reputation reports by introducing the use of trust to maintain a node's behavior in the reputation system. In [3], the authors proposed a trust index to measure the performance of nodes in reporting and locating events. They focus only on binary events and assume nodes are capable of individually reporting event locations instead of just a distance to the event radius. [4] presents an information theory model of representing and using trust in ad-hoc networks, where they lay out two ways of reasoning about trust and several axioms that can be used to draw conclusions based on both first-hand and second-hand observations.

III. Models

A. System model

The network comprises of static sensor nodes of identical capacity. Nodes will organize themselves into clusters (e.g., according to the Low-Energy Adaptive Clustering Hierarchy (LEACH) [8] protocol). Events of known amplitude can occur at any point within the area of the network. The events are sensed by nodes within a limited distance radius in two dimensions. The sensing range of each node is known. Data aggregation consists of the CH using the reported sensor measurements and the appropriate sensor model to detect the occurrence and location of an event.

A simple yet realistic model for sensing is given by

$$z_i = \frac{a}{\|\mathbf{x} - \mathbf{\xi_i}\|^{\alpha}} + w \qquad (1)$$

where z_i is the received sensor measurement, a is the amplitude of the signal, $\mathbf{x}$ is a 2-dimensional vector indicating the event position, ξ_i is a 2-dimensional vector indicating the location of sensor node i, α is a known attenuation coefficient, w is the additive white Gaussian noise and $\|x\|$ gives the Euclidean norm of x. Every node is aware of its location and the location of all other nodes in its cluster. Much work has gone into the area of node localization for sensor networks, both in secure environments [9] and insecure environments [10]. The existing work can be used to meet this assumption. RDAS is resilient to normal errors in the localization process. We assume the nodes will use some basic cryptographic mechanisms to authenticate their identity , e.g., when broadcasting reputation scores, and prevent Sybil attacks. We assume there exist out-of-band mechanisms to distinguish between events closely spaced in time so that our data aggregation protocol can determine which reports correspond to which event. RDAS works irrespective of relative values of transmission and sensing ranges.

B. Failure model

We assume a failure model with increasing levels of sophistication. We only consider compromised internal nodes since they have access to cryptographic keys. External nodes can be eliminated from data aggregation since they will not have valid keys. The process of compromising a node requires expenditure of non-zero amount of resources, including time. Therefore, at the time of deployment, there exists no compromised internal node and the network becomes compromised at a certain finite rate. A *legitimate node* will report the data that it senses and will commit errors according to a specific distribution with a small mean value. A *faulty* node will report incorrect sensor measurements with the goal of altering the result of data fusion. The error in these measurements will statistically be higher than that in the reports from legitimate nodes. The faulty behavior can be colluding or non-colluding. A *non-colluding faulty* node randomly sends incorrect information with no specific pattern, while a *colluding faulty* node conspires with other faulty nodes by coordinating the value of their erroneous reports. One example of this would be that they agree on a single faulty location and report a sensed value that corresponds to an event at that location.

Nodes may also mis-report on other nodes since second-hand information is used in our reputation system. A *bad-mouthing* node will falsely accuse legitimate nodes of incorrect behavior, while a *ballot-stuffing* node will falsely praise faulty nodes. Both kinds will be referred to as *liar* nodes. Nodes which exhibit both faulty and liar behavior will be referred to as *malicious nodes*. Malicious nodes can be smart and can behave correctly for some time to generate favorable ratings.

IV. Reputation-based Resilient Data Aggregation

The goal of our system is to reliably and efficiently determine the location of an event given sensor measurements in the presence of legitimate, faulty, and malicious nodes. To make the discussion concrete, we use LEACH as the underlying data aggregation protocol, though the principles will also apply to other data aggregation protocols. RDAS can be split into two separate yet tightly coupled components: (1) the reputation system and (2) the secure data aggregation protocol which performs aggregation in the presence of adversaries by using the reputation system.

A. Reputation System

The reputation engine adopted for our design is called the *beta reputation system* [7]. It has the advantages of having a firm basis on the theory of statistics and being intuitive in its application. The beta probability distribution is used to describe the posteriori probability of a binary event based on past observed outcomes of the event. Its probability density function can be described using the gamma function as:

$$f(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{2}$$

where, there are $\alpha + \beta$ past outcomes, p is the probability of occurrence of outcome α and $(1-p)$ the probability for occurrence of outcome β.

We follow a Bayesian framework using the beta distribution for representing reputation in the sensor network. We consider a node's actions to be binary events with the possible outcomes being correct or erroneous. Our goal is then to estimate the probability that the behavior of the node for the next event is correct. Node i believes that node j will behave correctly with probability θ. The outcome is independently drawn from observation to observation and θ is different for every node j. Since parameter θ is unknown, node i models this uncertainty by assuming θ is drawn from a beta distribution that is updated as new observations are made. We use the $Beta(\alpha, \beta)$ distribution to estimate θ, where α and β represent the count of correct and erroneous actions observed respectively. At the start of the system, without prior knowledge, the initial estimate of θ should correspond to a uniform distribution on [0,1], or equivalently Beta(1,1). As more observations are made, the beta probability density function asymptotically approximates a Dirac at θ. We define the reputation rating $R_{i,j}$ that node i has about node j as the expected value of the beta distribution parameterized as follows

$$R_{i,j} = E(Beta(\alpha+1, \beta+1)) = \frac{\alpha+1}{\alpha+\beta+2} \tag{3}$$

This formulation is intuitive because the reputation rating increases (decreases) as more correct (erroneous) actions are observed. Also, it gives an initial reputation of 0.5 indicating any action is equally likely from the node in the absence of any other knowledge.

1) Representing reputation and trust: Each node maintains its own reputation ratings for all other nodes with which it interacts. Nodes can behave arbitrarily in performing different classes of tasks. For example, a node can act correctly while reporting data to its CH but can shift the outcome of aggregation when performing CH duties. Reputation thus is represented by a vector $R_{i,j} < q_1, ..., q_n >$ with a dimension for each of the n different classes of tasks node i is observing node j perform. Reputation of a node can be built from direct observations of the node or from second-hand reports of the node's behavior. Second-hand information is desirable in order to confirm first-hand information and to update reputation faster, since nodes may only have fleeting interaction with some nodes. However, this also makes the system vulnerable to bad-mouthing and ballot-stuffing attacks.

We adopt the approach from [1] in dealing with liar nodes. We separately keep track of a node's accuracy in reporting on the behavior of other nodes and represent it in a metric called *trust rating*. The trust $T_{i,j}$ that node i has in node j is described by using the beta distribution

$$T_{i,j} = E(Beta(\gamma+1, \delta+1)) = \frac{\gamma+1}{\gamma+\delta+2} \tag{4}$$

where parameters γ, δ represent correct reporting and erroneous reporting actions, respectively. The trust rating can be used to decrease the influence of liar nodes in the reputation system by discounting second-hand reports according to a node's trust value.

2) Calculating reputation and trust: For a node i, RDAS calculates α, β as follows. Node i interacts with node j for some Δt and records $r_{i,j}$ cooperative events and $s_{i,j}$ non-cooperative events. It also receives a report from a set N of neighbor nodes with their own observations . These will all be aggregated as follows

$$\begin{aligned} \alpha_{i,j}^{new} &= u\alpha_{i,j} + r_{i,j} + \sum_{k \in N} D(r_{k,j}) \\ \beta_{i,j}^{new} &= u\beta_{i,j} + s_{i,j} + \sum_{k \in N} D(s_{k,j}) \end{aligned} \tag{5}$$

where $u < 1$ is an aging factor that allows reputation to fade with time. The last term incorporates second-hand information to the reputation calculation. The problem of how to combine first-hand and second-hand information into a single reputation measure was tackled by the proponents of the beta reputation system in [7] by mapping it to a similar problem in Dempster-Shafer belief theory [11]. We use the trust rating of k to weight down its second-hand report about node j to node i. The $D(r_{k,j})$ belief discounting function is defined as

$$\begin{aligned} D(r_{k,j}) &= \frac{\{2*\gamma_{i,k}*r_{k,j}\}}{\{(\delta_{i,k}+2)*(r_{k,j}+s_{k,j}+2)\}+\{2*\gamma_{i,k}\}} \\ D(s_{k,j}) &= \frac{\{2*\gamma_{i,k}*s_{k,j}\}}{\{(\delta_{i,k}+2)*(r_{k,j}+s_{k,j}+2)\}+\{2*\gamma_{i,k}\}} \end{aligned} \tag{6}$$

A detailed mathematical explanation is in [12]. This function gives greater weight to nodes with high trust and never gives a weight above 1. Therefore second-hand information will never outweigh first-hand information. If $\gamma_{i,k} = 0$ the function will return 0, therefore node k's report will not affect the reputation update. The relationship between trust and weight is not linear, as the weight value is slightly less than trust value.

In order to update the trust rating, a node i receiving a reputation report from node k about node j compares this report to its own reputation rating for node j. If the difference exceeds a threshold then node i lowers its trust rating for node k. Let a be the result of the following deviation test

$$a = \begin{cases} 1, & |R_{i,j} - E((Beta(r_{k,j}, s_{k,j})))| < d \\ 0, & |R_{i,j} - E((Beta(r_{k,j}, s_{k,j})))| \geq d \end{cases} \tag{7}$$

where d is a positive constant for the deviation threshold. Here $a = 1$ corresponds to what the node perceives as legitimate behavior. So after the deviation test, we update our trust:

$$\gamma := v\gamma + a; \delta := v\delta + (1-a) \tag{8}$$

where $v < 1$ is the aging factor for trust fading. The idea is that a node k that is lying about some node j in its reports will see its trust rating reduced, since under the assumption that a majority of the nodes are correctly reporting the behavior of node j, $R_{i,j}$ will be different than the reports from node k. If node j's behavior is changing, then the trust of node k will suffer initially, but will go back up when its reports are confirmed by other neighbors of node i and i itself.

3) Generating reputation: A critical decision in the design of a reputation system is how to generate reputation. In order to accurately quantify reputation we must be able to distinguish between malicious behavior and natural errors in the sensor network. Under the natural assumption that the majority of the nodes in the network will not be compromised, majority voting seems to be the most natural way to distinguish between correct and incorrect data. However, as the nodes are located at different distances from the event, their sensed values will be different. RDAS introduces the use of CHs to generate reputation during the data aggregation phase. Every node within a cluster sends their sensed data to the CH at fixed time intervals. In order to obtain a reliable indication of how accurate a sensor's data is, the CH uses the result of data aggregation to determine how data from one node compares to the data reported by the rest of the cluster nodes (CNs).

Data aggregation in our system model consists of the detection and localization of events. After the initial cluster formation, nodes are assigned a time slot for data transmission to the CH. If a node sensed an event since the last data transmission cycle, it transmits the sensed value at its allocated time slot, following the LEACH protocol [8]. At the end of data transmission the CH determines that an event occurred based on the sensed values and weighting a node's report (or silence) by its reputation value. If the CH determines an event occurred then it estimates the location of the event using redundant data obtained from the CNs and calculates what sensor value should have been reported by every individual node. This value can be compared to the actual value the sensor node reported in order to determine the accuracy of the report. This process is illustrated in Figure 1.

In order to measure the accuracy of an individual node's report, the CH uses Equation 1 and solves for z_i (neglecting the w noise term) to determine the expected sensor measurement for node i. By analyzing the individual error of each node the CH can determine which nodes' measurements fall within the expected error range and which deviate from that range. Error ranges for individual nodes will depend on many variables including sensor type, event type and distance to the event and therefore, we do not make any such assumption in our model. The following formula is used to determine whether node i's data report, denoted as z_i, is accurate or not

$$X_i = \frac{|\hat{z_i} - z_i|}{\hat{z_i}} \tag{9}$$

$$a = \begin{cases} 1, \ X_i - \mu < k\sigma \\ 0, \ otherwise \end{cases} \tag{10}$$

where $\hat{z}_i$ is the sensor measurement estimate obtained by the CH using Equation 1, μ and σ are the average and standard deviation, respectively, of X_i across all CNs. The rationale for this is that legitimate nodes will have on average a similar error range in their sensing. Therefore nodes whose error is considerably larger than the average error range will be deemed faulty. Setting a to 0 denotes the conclusion that the activity is erroneous. Equation 10 applies the one-sided Chebyshev's inequality by treating X_i as a random variable with mean μ and by setting a parameter k such that the probability that a cooperative node's sensor measurement is greater than $k\sigma$ is less than or equal to $\frac{1}{1+k^2}$. By choosing the parameter k, the system owner can bound the probability of location error exceeding a certain threshold ($\mu + k\sigma$).

If an event was detected and a node did not report any data then the CH calculates the distance between the node and the event and compares it to the sensing range of the node's sensor. If the distance is less than the sensing range by a fixed tolerance the CH sets $a = 0$, otherwise $a = 1$. Similarly, if an event was not detected, the CH sets $a = 0$ to nodes which reported data, and $a = 1$ to those who did not. Reputation reports are then generated as follows

$$r_i := ur_i + a; s_i := us_i + (1 - a) \tag{11}$$

where r_i and s_i are the number of cooperative and non-cooperative events, respectively, observed about node i during a given period of time. This calculation is done at the CH. So the CH updates the reputation scores of all the other nodes in the cluster. The CH sends the tuple (r_i, s_i) for each node in a broadcast to all the nodes in the cluster. But, it does not send its reputation score for any node in the broadcast. CNs will update their reputation from this report using Equation 3. Thus, a CN only obtains second hand information about the other CNs in the cluster through the CH.

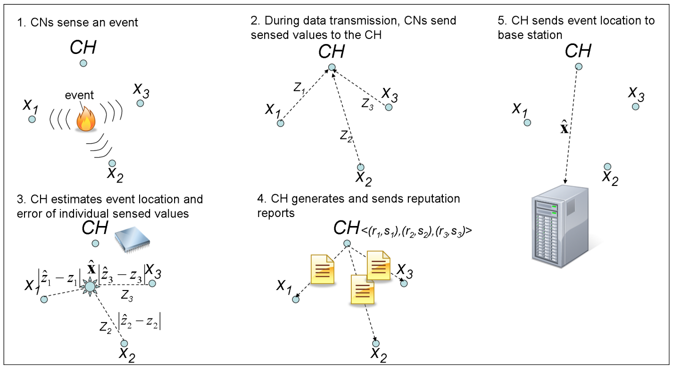

1. CNs sense an event
CH
event
x_1
x_3
x_2
2. During data transmission, CNs send sensed values to the CH
CH
z_1
z_3
z_2
x_1
x_3
x_2
5. CH sends event location to base station
CH
$\hat{\mathbf{x}}$
x_1
x_3
x_2
3. CH estimates event location and error of individual sensed values
CH
$\hat{\mathbf{x}}$
x_1
$|\hat{z}_1 - z_1|$
$|\hat{z}_3 - z_3|$
x_3
z_3
z_2
$|\hat{z}_2 - z_2|$
x_2
4. CH generates and sends reputation reports
CH<(r1,s1),(r2,s2),(r3,s3)>
x_1
x_3
x_2

Fig. 1. Reputation generation during data aggregation

B. Data Aggregation

The second aspect of our system design deals with the effective use of reputation in order to increase the reliability of data aggregation in the face of malicious nodes and legitimate nodes making natural errors. We exemplify the data aggregation with event localization, though the principles apply to any data aggregation where overlapping data reports are available from multiple nodes.

1) Resilient Event Localization: The CH determines if an event has occurred by using majority voting among the CNs in its cluster. In order to prevent faulty nodes from shifting the result of the voting, RDAS uses weighted voting at the CH to reduce the influence of nodes with low reputation. For a cluster C, every node j that reports sensor data submits a vote $e_j = 1$, and every node j that does not report data submits a vote $e_j = 0$. Every vote is weighted down by the reputation of node j and counted as follows

$$\begin{aligned} EventVotes &= \sum_{j \in C} R_{i,j} * e_j \\ NoEventVotes &= \sum_{j \in C} R_{i,j} * (1 - e_j) \\ Event &= \begin{cases} 1, \ EventVotes > NoEventVotes \\ 0, \ otherwise \end{cases} \end{aligned} \tag{12}$$

where $Event = 1$ represents that an event has been detected.

The goal of the CH is to eliminate the contribution of faulty nodes from the data aggregation. There are two alternate approaches that two variants of RDAS use. The CH applies reputation filtering, which consists of eliminating nodes whose reputation drops below a filtering threshold (FT) from the event localization aggregation. A second approach, called reputation weighting, is to apply weighted least-squares fitting using a function of the nodes' reputation rating as the weight for each equation. Weighted least-squares fitting gives priority to minimizing the error of equations with a greater weight. Nodes with a higher reputation can thus have more influence on determined location. Assigning high reputation weights to well-behaving nodes allows RDAS to work despite a majority of nodes being compromised. See [12] for the equation of resilient event localization using reputation weighting.

2) Cluster head election and monitoring: So far all reputation generation, propagation and event localization have been performed by the CH. Thus, there is nothing to prevent the CH from performing malicious activities. An unmonitored CH can omit sending the event location or send false data to the base station, and can send false reputation reports about nodes in its cluster. To address this, we use *cluster monitors* (CMs) which perform the same data aggregation as the CH. But instead of sending a high-energy transmission to the base station, they overhear the CH's report to the base station, compare it with the result of their own aggregation and generate a reputation report about the CH. A reputation rating about CH behavior will be kept as a dimension of the reputation vector, since it can be assumed that nodes could behave correctly as sensing nodes and maliciously as cluster heads. If the CH has low reputation, its broadcast can be disregarded by all the CNs. Also, the CH will not be elected again.

In RDAS, nodes in a round become CMs with twice the probability of becoming a CH, so there will be on the average two CMs per cluster. In the first communication phase of the LEACH protocol, the CHs broadcast an advertisement as do the CMs. The CNs choose the maximum signal strength cluster after discarding advertisements from nodes whose reputation is below a specified threshold. Cluster monitors will compare their aggregation with the report the CH sends to the base station, and generate a reputation report depending on the relative difference in aggregation values as follows

$$\begin{aligned} r_{i,j} &= \begin{cases} 1, \ \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\| < AggThr \\ 0, \ otherwise \end{cases} \\ s_{i,j} &= \begin{cases} 1, \ \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\| \geq AggThr \\ 0, \ otherwise \end{cases} \end{aligned} \tag{13}$$

where $\hat{\mathbf{x_i}}, \hat{\mathbf{x_j}}$ is the estimated event location by nodes i and j respectively. Three legitimate nodes may arrive at slightly different results due to various sources of inexactness in the scheme, such as imperfect detection in the wireless environment and inaccuracies in location estimation. This leads us to use the threshold for comparison. Cluster monitors will generate reputation reports about regular CNs in the same fashion as the CHs. This makes them serve a second purpose in accelerating the propagation of reputation across the network.

V. Experiments and Results

We simulated the data aggregation protocol using the TinyOS 2.0 Simulator (TOSSIM). Nodes in the sensor network are placed in uniformly distributed random locations within a square area. We simulate the inaccuracy of location estimate by making each node add a normally distributed random error to the location of every node. An event of fixed amplitude is generated at fixed intervals at a different random location. Nodes that are within sensing range of the event detect its occurrence during their sensing period and report the sensed value following the data aggregation protocol. The compromised nodes in the network perform some of the faulty actions detailed in Section III-B with a fixed probability. The Filtering Threshold (FT) approach is used at the CH to filter out input from nodes with low reputation ratings. We separately measure the propagation of reputation across the network over time and the accuracy of reputation-based data aggregation. The accuracy of data aggregation is quantified by the distance between the event location estimate at the CHs and the actual location of the event. The metric we are specifically interested in is the improvement in event localization compared to the baseline data aggregation without any reputation or trust. We call this metric the *accuracy* of our system, defined as:

TABLE I
Network & RDAS Parameters

Parameter	Value
Area of sensor field	300m by 300m
Node transmission range	100 m
Node sensing range	100 m
# Nodes	50
# Clusters	3
Radio propagation model	Shadowing model
k Constant in Chebyshev's inequality	1
Filtering Threshold (FT)	0.5
Aging Factor	0.99
Trust Deviation Threshold	0.3

$$Accuracy = 1 - \frac{Localization\ error\ using\ RDAS}{Localization\ error\ using\ baseline} \tag{14}$$

An accuracy value greater than zero indicates RDAS outperforms the baseline. Accuracy is plotted against events rather than time, since reputation and trust are updated only when events occur. Experiments were repeated 10 times to obtain low confidence intervals. The same set of events is used for computing the accuracy in RDAS and the baseline.

A. Experiment 1 - Effect of Faulty Nodes

In Experiment 1, we analyze the effect of increasing compromise rates on the breaking point of the system. We linearly increase the percentage of compromised nodes in increments of 6%, starting at zero and increasing up to 72%. After more than 72% the results become too unstable to be useful. In this experiment compromised nodes are faulty, meaning they report erroneous sensor measurements. For part (a), the faulty nodes are non-colluding and add a uniform random number to their sensed data values. For part (b), the faulty nodes are colluding and report data corresponding to an event location which they have previously coordinated and which is different from the actual event's location. In a real world scenario nodes must be compromised at some finite rate, so we repeated the experiment for different rates.

1) Experiment 1(a) - Non-colluding faulty nodes: Figure 2(b) shows the accuracy of the system as the network gets compromised at two different rates of 6% every 400 and 6% every 800 events. Each marker in the plots indicates the point at which an additional 6% of the network became compromised. Figure 2(c) zooms in on the first 4000 events. It is interesting to observe that during the first 400 events, when no nodes are compromised, the accuracy drops below 0. This is due to the fact that the reputation threshold used for filtering is 0.5 and all nodes start with a rating of 0.5. Initially some nodes have their reputation reduced due to natural errors, and their data is not used for event localization.

After the first nodes become compromised the localization error of the both the baseline and RDAS systems increases equivalently, but as compromised nodes start gathering bad reputation the error of RDAS becomes less than that of the baseline system and the accuracy of the system increases with each passing event. When previously legitimate nodes become compromised, the accuracy of the system drops for a short period since these nodes have built high reputation ratings before becoming compromised and their faulty data will be aggregated by CHs. Thus the reputation ratings of the compromised nodes spike when fresh nodes are compromised (Figure 2(d)). Conversely, the reputation of legitimate nodes suffers slight drops. This explains the oscillating pattern of the accuracy plots immediately after additional nodes become compromised. The two different rates indicate that the faster the network gets compromised the harder it is to maintain

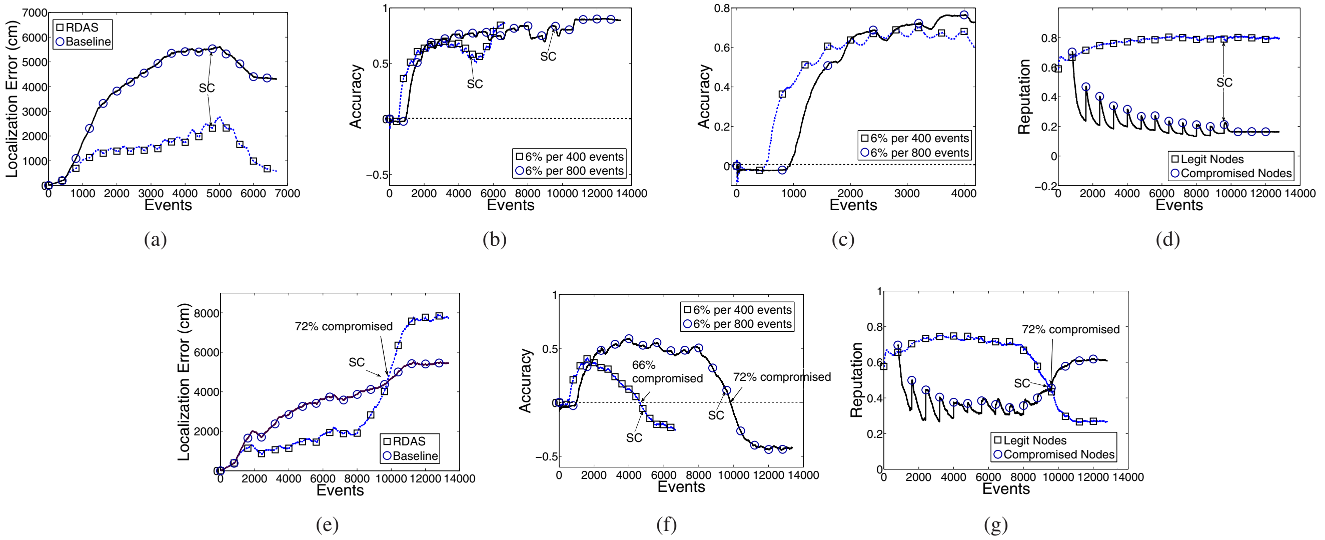


Fig. 2. Localization error, accuracy, average reputation and average trust as network is compromised at a rate of 6% every (a) 400 events and (d, e, g) 800 events. The first and second rows are for non-colluding and colluding faulty nodes, respectively. (SC=stopped compromising).

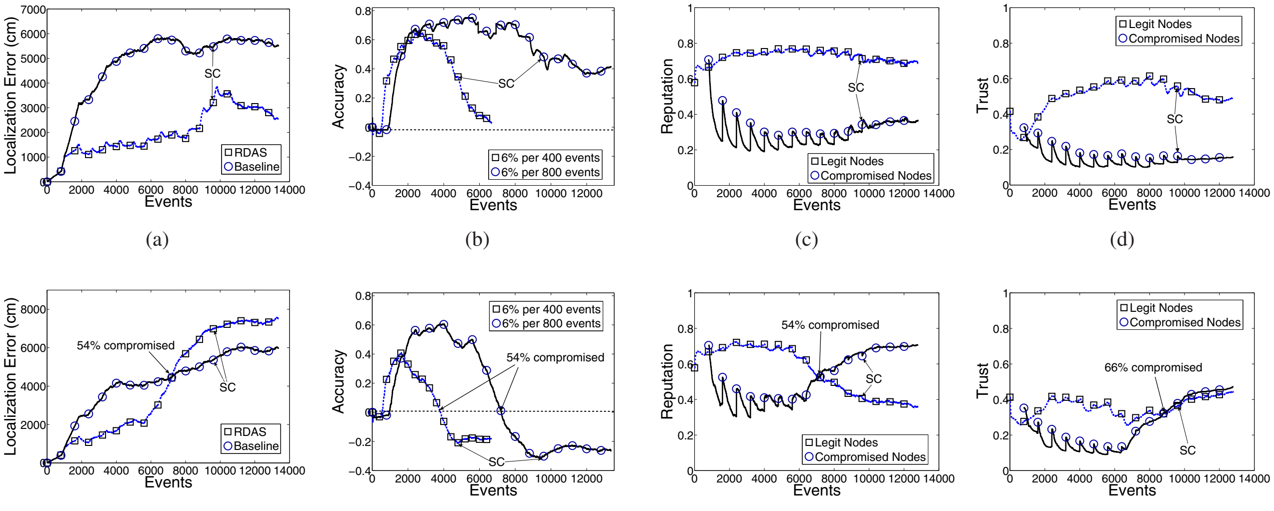
accuracy, since the nodes will have less time to detect compromised nodes through the reputation mechanism. In order to reduce the accuracy of RDAS an attacker would not only need to compromise a large number of nodes, but must do so fast enough so their combined influence will prevent their reputation decreasing relative to that of legitimate nodes. Figure 2(b) shows that the system with a 6% per 400 events compromise rate has 72% of its nodes compromised its accuracy is decreasing, but with no more nodes getting compromised after this point its accuracy increases sharply. The results indicate that even after 72% of the network has been compromised the system is able to maintain better accuracy than baseline and eliminate most of the false reports from non-colluding faulty nodes. Figure 2(a) illustrates that RDAS always has higher localization accuracy than the baseline with faulty nodes.

2) Experiment 1(b) - Colluding faulty nodes: The second part of Experiment 1 was done with the same rates of 6% per 400 events and 6% per 800 events, but with compromised nodes that collude in their misbehavior. The initial effect of colluding faulty nodes in the network is the same as the effect of non-colluding faulty nodes. But since reputation is generated by comparing data measurements among different nodes, as the number of colluding faulty nodes per cluster increases they will obtain high reputation ratings and the reputation of legitimate nodes will decrease.

Figure 2(f) shows how even after 54% of the network is compromised RDAS is able to maintain the accuracy of the system steadily above 0 following a faulty node injection. But after 60% of the network becomes compromised the accuracy slopes sharply down without recuperating. Since the faulty nodes are colluding, when faulty nodes form a majority within a cluster the result of event localization will be closer to their false event location than the real event location. Therefore the individual sensing error of the faulty nodes will be less than the average and they will get good reputation reports while the legitimate nodes get bad reputation reports. Eventually the colluding nodes' reputations will be higher than that of legitimate nodes and RDAS will perform worse than the baseline system. Figure 2(e) and Figure 2(g) illustrate the crossover point where RDAS begins to perform worse than the baseline system. For the regions where RDAS performs worse than the baseline, the localization error is high enough that neither scheme will be useful in practice. Note that RDAS is vulnerable to a plausible adversary model, in which nodes are compromised over time. However, they behave legitimately till more than half the nodes are compromised. Then they all collude and behave maliciously. To the best of our knowledge, no reputation scheme, wireline or wireless, can thwart this model.

B. Experiment 2 - Effect of Malicious Nodes

1) Experiment 2(a) - Non-colluding malicious nodes: Figure 3(b) shows that RDAS still performs better than the baseline system even as 72% of the network is compromised with malicious nodes. However, the addition of liar behavior reduces the gap in reputation ratings between compromised and legitimate nodes and therefore the accuracy of the system is less than with nodes that are just faulty (Experiment 1(a)). Figure 3(d) shows that initially with a small part of the network compromised RDAS is able to identify liar nodes to reduce their influence on the reputation system (since they have low trust values). The reputation of legitimate nodes therefore keeps increasing (Figure 3(b)), up until 60% of the nodes become compromised. Beyond that, the legitimate nodes' reputation starts to drop and the malicious nodes' reputation starts to rise. This causes the same effect on the trust ratings, since legitimate reputation reports will deviate from current reputation ratings and vice versa with reputation reports from liar nodes. This affects the system accuracy but not enough to



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 3. Localization error, accuracy, average reputation and average trust as network is compromised at a rate of 6% every 800 events (a, c, d, e, g, h). The first and second rows are for non-colluding and colluding malicious nodes, respectively. (SC=stopped compromising).

take it below 0, because the average reputations of legitimate and malicious nodes never cross for non-colluding malicious nodes. This is because the sensed data values the malicious nodes report do not agree well with the other reports.

2) Experiment 2(b) - Colluding malicious nodes: Colluding nodes affect the accuracy of the system by shifting reputation ratings in their favor as more nodes become compromised. Figure 3(g) shows that as the network gets compromised with colluding malicious nodes the average reputation of the compromised nodes will begin to rise above the reputation of legitimate nodes causing RDAS to perform worse than baseline (Figure 3(f)). This occurs after 54% of the network is compromised, so even with a little more than half of the network compromised RDAS is able to perform better than the baseline system. Figure 3(h) shows that the average trust of liar nodes exceeds the average trust of legitimate nodes after 66% of the network is compromised. The system is therefore unable to identify and repress liar nodes after this point. However, beyond the 54% mark, neither system is usable due to high absolute localization error. We also performed experiments where a fraction of the nodes has been secured and cannot be compromised. The network gets compromised at a constant rate until it reaches 30% and with all compromised nodes colluding. Here RDAS always outperforms the baseline — when nodes are being compromised, and even more significantly after node compromise stops. Once the network stops getting compromised further, RDAS causes the accuracy to improve linearly over time. The detailed results can be found in [12].

VI. Detection Probability and Overhead Analysis

In this section we present an analytical estimate of the probability of detecting faulty nodes as a function of their number in the network and behavior. Given a set of nodes C that form a cluster, nodes within this set will form two partitions M and $\bar{M}$, of faulty and legitimate nodes respectively. Our goal is to classify a node's behavior during a data aggregation round as legitimate or faulty. This is achieved by computing the relative error X_i described in equation 9 and using the deviation test in equation 10. The distribution of this error will depend on the fault model of the given node. For any node in C we can describe the probability density function of the error X using the theorem of total probability:

$$f_X(x) = m f_X(x|M) + (1-m) f_X(x|\bar{M}) \tag{15}$$

where m is the probability that a node is faulty. Assume faulty nodes follow a distribution with mean μ_M and variance σ_M^2, while legitimate nodes follow a distribution with mean $\mu_{\bar{M}}$ and variance $\sigma_{\bar{M}}^2$. Although the exact distribution of X depends on the fault model of the given node, the expected value of the error for legitimate nodes, μ_M, can be reasonably assumed to be zero. Reasonably $\mu_M > \mu_{\bar{M}}$ and $\sigma_M = \sigma_{\bar{M}}$ for colluding nodes and $\sigma_M > \sigma_{\bar{M}}$ for the non-colluding nodes. The expected value and variance of X will then be given by:

$$E(X) = m\mu_M + (1-m)\mu_{\bar{M}} \tag{16}$$

$$\begin{aligned} Var(X) &= E(X^2) - E^2(X) \\ &= m(\mu_M^2 + \sigma_M^2) + (1-m)(\mu_{\bar{M}}^2 + \sigma_{\bar{M}}^2) \\ &\quad -m\mu_M^2 - 2m(1-m)\mu_M\mu_{\bar{M}} \\ &\quad -(1-m)^2\mu_{\bar{M}}^2 \end{aligned} \tag{17}$$

Given data measurements from a cluster, the probability that a node i's measurement will be classified as faulty is given by (from Equation 10):

$$P(detection) = P(X \geq \sigma_X k + \mu_X) \tag{18}$$

For $i \in M$ we can calculate this probability using the conditional cumulative density function $F_X(x|M)$.

$$P(detection) = 1 - F_X(\sigma_X k + \mu_X | M) \tag{19}$$

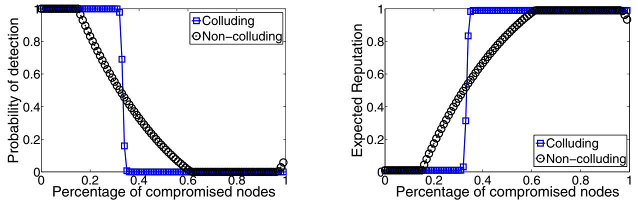
Figure 4(a) shows the probability for these two fault models as the fraction of faulty nodes, namely m, increases. For this graph, the model used for legitimate nodes is $N(\mu_{\bar{M}} = 0, \sigma_{\bar{M}} = 0.03)$. Non-colluding faulty nodes follow model $U(2,4)$ which corresponds to $\mu_M = 3, \sigma_M = 0.57$, while colluding faulty nodes follow model $N(\mu_M = 3, \sigma_M = 0.03)$. The plot shows that for a small fraction of faulty nodes the scheme should be able to detect faulty behavior with high probability for both fault models. For non-colluding faulty nodes, the uniformly random noise added to their data measurements causes the probability of their detection to drop gradually. For colluding faulty nodes, their coordinated data reports cause the probability of detection to drop drastically when 35% of the network is compromised.

This analysis provides a worst case estimate of the performance of RDAS. This is due to the fact that the analysis emulates the case where the entire fraction of compromised nodes (given by the value on the x-axis) is introduced in one go, where each faulty node has the same reputation rating as the legitimate nodes. The network is then executed for an extended period of time and asymptotically the output metrics (reputation or probability of detection) approach the value given on the y-axis. In practice however, nodes get compromised gradually, which will lead to better performance.

Using this model, we can estimate the steady state reputation of a compromised node as seen by a legitimate node after a fixed number of aggregations. Every aggregation can be seen as a Bernoulli trial, with probability of success $p_\alpha = 1 - P(detection)$ and $p_\beta = P(detection)$. After n trials, the expected values are $E(\alpha) = np_\alpha$ and $E(\beta) = np_\beta$. The expected value of reputation is then given by

$$E(Rep) \quad = \quad \frac{np_\alpha + 1}{np_\alpha + np_\beta + 2} \tag{20}$$

For the same scenario case of non-colluding and colluding faulty nodes illustrated in Figure 4(a), the expected reputation is plotted in Figure 4(b) (P(detection) is obtained from previous analysis, n = 100).



(a)

(b)

Fig. 4. (a) Probability of detection and (b) expected reputation of faulty node as fraction of faulty node increases.

Overhead Analysis. The overhead due to RDAS for bandwidth is only for broadcast of reputation reports by the CH to all the CNs in the cluster. For a C node cluster and A actions by a node in a round, the amount of overhead data per round in bits is $BW_{rep} = C * (\lceil log_2(C) \rceil + \lceil log_2(A) \rceil)$. For a 20 node cluster (as suggested by LEACH [8]) and 100 actions, this only comes to 30 bytes. For the memory overhead, it is only for storing values of α, β, γ, and δ (1 byte each) for each node within twice the transmission range. For a transmission range R and network node density D, the memory consumption is $MC = 4\pi R^2 * D * 4$ bytes. For the case of a network with uniformly distributed nodes with density 0.0025 nodes per m^2 and transmission range of 100 m, $MC = 502$ bytes.

VII. Conclusions

Here we described RDAS, a robust data aggregation system that tolerates unreliable and adversarial nodes by maintaining reputation ratings to detect nodes that report faulty data. The cluster head uses the ratings to prevent erroneous data from affecting the aggregation result. Simulation results show that for a network that is becoming increasingly compromised, RDAS is able to quickly detect nodes that are reporting misleading data and eliminate their influence on event localization. Though we discussed RDAS using localization as the underlying aggregation, our scheme can also be used for other aggregation operations. RDAS assumes some hierarchical structure in the network (like cluster head, cluster nodes etc.) and redundancy in the data reports such that the same event is reported by multiple nodes. Higher level aggregation operations, such as SUM and COUNT, can be built on top of this. For example, by using reputation filtering, the CH can calculate the COUNT of how many nodes reported an event and how many did not. As part of future work, we plan to extend RDAS to multi-hop routes. This adds security concerns due to the possibility of nodes dropping or modifying data enroute to the destination.

References

[1] S. Buchegger and J. L. Boudec, “Robust reputation system for p2p and mobile ad-hoc networks,” in *Economics of Peer-to-Peer Systems*, 2004.

[2] S. Ganeriwal and M. Srivastava, “Reputation-based framework for high integrity sensor networks,” in *SASN*, 2004.

[3] M. Krasniewski, P. Varadharajan, B. Rabeler, S. Bagchi, and Y. Hu, “TIBFIT: Trust index based fault tolerance for arbitrary data faults in sensor networks,” in *DSN*, 2005.

[4] K. Yan Lindsay Sun; Wei Yu; Zhu Han; Liu, “Information theoretic framework of trust modeling and evaluation for ad hoc networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 2, pp. 305–317, 2006.

[5] B. Przydatek, D. Song, and A. Perrig, “Sia: Secure information aggregation in sensor networks,” in *SenSys*, 2003.

[6] W. Zhang, S. Das, and Y. Liu, “A trust based framework for secure data aggregation in wireless sensor networks,” *IEEE SECON’06*, 2006.

[7] A. Josang and R. Ismail, “The beta reputation system,” in *15th Bled Electronic Commerce Conference*, 2002.

[8] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *HICSS*, 2000.

[9] D. Moore, J. Leonard, D. Rus, and S. Teller, “Robust distributed network localization with noisy range measurements,” in *SenSys*, 2004, pp. 50–61.

[10] L. Lazos and R. Poovendran, “Serloc: Robust localization for wireless sensor networks,” *ACM Trans. Sen. Netw.*, vol. 1(1), pp. 73–100, 2005.

[11] A. Josang, “A logic for uncertain probabilities,” *Intl Jrnl of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 3, 2001.

[12] C. Perez, “Reputation-based Resilient Data Aggregation in Sensor Network,” Purdue Masters Thesis, pp. 1-60, at http://http://docs.lib.purdue.edu/ecetheses/11/, 2007.