Reputation-based Framework for High Integrity Sensor Networks

Saurabh Ganeriwal and Mani B. Srivastava
Networked and Embedded Systems lab
56-125B, EE-IV, University of California Los Angeles
{saurabh, mani}@ee.ucla.edu

ABSTRACT

The traditional approach of providing network security has been to borrow tools from cryptography and authentication. However, we argue that the conventional view of security based on cryptography alone is not sufficient for the unique characteristics and novel misbehaviors encountered in sensor networks. Fundamental to this is the observation that cryptography cannot prevent malicious or non-malicious insertion of data from internal adversaries or faulty nodes.

We believe that in general tools from different domains such as economics, statistics and data analysis will have to be combined with cryptography for the development of trustworthy sensor networks. Following this approach, we propose a reputation-based framework for sensor networks where nodes maintain reputation for other nodes and use it to evaluate their trustworthiness. We will show that this framework provides a scalable, diverse and a generalized approach for countering all types of misbehavior resulting from malicious and faulty nodes.

We are currently developing a system within this framework where we employ a Bayesian formulation, specifically a beta reputation system, for reputation representation, updates and integration. We will explain the reasoning behind our design choices, analyzing their pros & cons. We conclude the paper by verifying the efficacy of this system through some preliminary simulation results.

Categories and Subject Descriptors

C.2.2 [**Computer Systems Organization**]: Computer Communication Networks – *Network Protocols.*

General Terms

Design, Reliability, Security.

Keywords

Sensor Networks, Security, Reputation, Trust, Cryptography, Framework, Bayesian Formulation.

SASN'04, October 25, 2004, Washington, D.C., USA.

1. INTRODUCTION

Security breach can happen in a sensor network not only while relaying information to the end-user but also while generating information. The ability of a sensor network to perform its task depends not only on its ability to communicate among the nodes, but also on its ability to sense the physical environment and collectively process the sensed data. This decentralized in-network decision-making, which relies on the inherent trust among the nodes, can be abused by adversaries to carry out security breaches through compromised nodes. Note that sensor nodes are envisioned to be low-cost which make it infeasible for manufactures to make them tamper-resistant; an adversary can undetectably take the control of a sensor node by physically compromising it. An adversary can then potentially insert faulty data or decisions to mislead the whole network! Cryptographic and authentication[1] mechanisms alone cannot be used to solve this problem as internal adversarial nodes will have access to valid cryptographic keys.

Besides malicious attacks, sensor nodes are also vulnerable to system faults. Non-malicious behavior such as malfunctioning of radio/sensors can also result in the generation of bogus data, bringing equally detrimental effects to the functioning of the network. The very nature of this type of misbehavior is outside the realm of cryptography.

In this paper, we make a case for integrating tools from different domains such as economics, statistics and data analysis with cryptography for developing trustworthy sensor networks. Based on this observation, we take an approach motivated from existing human societies in the world. Embedded in every social network is a web of *trust;* with a link representing the trustworthiness between two individuals. When faced with uncertainty, individuals seek the opinions of those they trust. The intent is to develop a similar Reputation-based Framework for Sensor Networks (RFSN), where sensor nodes maintain *reputation* for other nodes. This reputation is used to evaluate the trustworthiness of other nodes. This establishes a web of trust in the network, which is then used as an inherent aspect in predicting the future behavior of nodes in the network. As we will show, RFSN not only provides a unified approach for countering all types of malicious/non-malicious misbehavior but also provides a framework for integrating different security solutions.

We are currently architecting a system, based on RFSN that can run on resource constrained sensor nodes such as Berkeley Motes. In this system we employ a Bayesian formulation, specifically

[1] Referred as cryptography throughout the paper.

beta reputation system, for reputation representation, updates, integration and trust evolution. We will provide a thorough analysis of our various system design choices, contrasting them with existing reputation systems. As a proof of concept, we will conclude the paper by presenting some preliminary simulation results.

2. RELATED WORK

In absence of adequate security, deployment of many applications of sensor networks could be curtailed. Only recently researchers have started looking into this matter. The testimonies to this are some recent works in literature that study the impact of malicious attacks on these networks [1, 2, 3]. There have been several proposals, all based on cryptography, to ensure secure communication on these resource constrained nodes such as SPINS[4], TinySec[5], INSENS[6], TinyPK[7], SERP[8], SEF[9] etc. The establishment and management of cryptographic keys [4, 8, 10, 11, 12] form the backbone of these schemes; the scale and ad-hoc deployment of nodes coupled with the ability of adversaries to easily recover the cryptographic materials make it a challenging problem to solve. The focus of this paper is different, and is based on the observation that conventional view of security based on cryptography alone is not sufficient for the novel misbehaviors encountered in sensor networks.

Trust-management approach for distributed systems security [13] was first introduced in the context of Internet as an answer to the inadequacy of traditional cryptographic mechanisms. Some of the notable earlier works in this domain have been trust-management engines such as KeyNote [14] and RT framework [15]. Since then, reputation-based frameworks based on the approach of trust management have been extensively studied in many contexts and equally diverse domains such as human social networks, e-commerce, 802.11 networks, peer-to-peer networks etc. In this paper, we study the applicability of this approach in developing high integrity sensor networks.

RFSN does borrow some design features from several existing works in literature but as a complete system differs from all the existing reputation-based systems such as KeyNote [14], RT framework [15], eBay [16], confidant [17], core [18], peer-to-peer networks [19] etc. They are either limited in scope (e.g. confidant and core focus on countering routing selfish misbehavior attacks) or differ in architecture (e.g. centralized model of eBay as opposed to a distributed architecture of RFSN). Moreover, none of these systems has been tested on real embedded systems and either assume a deterministic model for representing reputation [16] or portray a very high level picture of the probabilistic framework based on trivial and debatable heuristics [14, 15, 17, 18, 19]. On the contrary, we employ a concrete Bayesian formulation, having strong foundation in statistics, for representing reputation and trust evolution. The most promising effort in this direction was carried out in [20] and [21], where authors propose to use a Bayesian formulation for representing reputation of a node in ad-hoc wireless networks although it was never fully developed.

It is well known that reputation systems can be tricked by the spread of false reputation ratings, be it false accusations or false praise [22]. In [21], a new promising approach of maintaining reputation of the reputation ratings have been proposed to counter these attacks, although it is far from being fully developed. We shall explain in Section 6 how we counter these attacks without making this explicit distinction between the performances in the base system from the one in the reputation system. Our approach implicitly takes into account the reputation of the reporter node during the integration step, thus fundamentally being similar to [21] and simultaneously saving the overhead of maintaining multiple reputation ratings for a node.

Further, we give special attention to maintaining a modular architecture of the framework, which is paramount to the development of any middleware solution for sensor networks. Our final goal is to show a running system on sensor nodes such as Berkeley motes.

3. LIMITATIONS OF CRYPTOGRAPHY

Cryptography presents an efficient mechanism for providing data confidentiality, data integrity, node authentication, secure routing and access control. We note that all these are definitely needed for developing secure sensor networks. However, we argue that cryptography alone is not sufficient for the unique characteristics and novel misbehaviors encountered in these networks. In this section, we highlight some of these scenarios to make a case for moving beyond the realm of cryptography to develop trustworthy sensor networks.

3.1 Collaborative Data Processing

Instead of providing a raw dump of sensor data, sensor networks often use in-network processing algorithms (aggregation) that besides saving energy also provide meaningful results to the end user. Similarly, sense-response applications such as fire monitoring and target tracking rely on decentralized decision making by a population of nodes. An inherent assumption is that all nodes will abide by the rules of the protocol. However, sensor nodes are envisioned to be cheap and therefore unlikely to be equipped with tamper-proof hardware. This coupled with the unattended operation of these networks leaves the node at the mercy of an adversary who can potentially steal nodes, recover their cryptographic material, and pose as an authorized node in the network. Thereafter, these internal adversaries can exploit the inherent trust between the nodes to abuse the functioning of these protocols.

To emphasize our point we present an example in Figure 1, depicting a sensor network deployed for intrusion detection. When a target is detected at *(x, y)*, the normal behavior of the network will be the following – Node *C* will collect the raw sensor data from *A* and *B*, it will fuse this information with its own reading to find the target's location and will eventually report this location to the end user using a multihop route through *D*, *E*, *F* and *G*. If an attacker compromises *C*, it can either hide the identity of the intrusion by not sending the processed results or can even mislead the user by reporting a false location estimate of the intruder, *(w, z)* instead of the real position *(x, y)*.

Only recently some proposals have been proposed to counter or restrict the impact of these attacks, SERP [8], SEF [9], SIA [23] etc., which uses the scale and redundancy in the system to their advantage. In general, cryptography alone cannot solve this problem as internally compromised nodes can generate bogus information and still authenticate it using valid cryptographic keys.

3.2 Data Authentication

Sensor network applications not only rely on the ability of nodes to communicate among themselves but also on their ability to sense the physical environment. However, internal adversaries after compromising a node (or its transducer) can insert bogus data, thereby misleading the whole network. For example, if an attacker compromises *A* or *B* (Figure 1), it can send false information "Target detected at *(w, z)*"; thereby making it hard for *C* to conclude anything about the intruder location. Note that data authentication is different from data integrity. Attaching message authentication codes can verify the consistency of data but cannot verify its validity as the source generating the data itself can be malicious. Cryptographic solutions will be again limited by the fact that adversaries have access to valid keys.

Besides malicious security breaches, bogus data can also be generated by nodes unintentionally due to the failure of some system components such as radio/sensors etc. Sensor nodes are made of cheap hardware components which are highly vulnerable to system malfunctioning and hence, such hardware failure are not uncommon in the realm of sensor networks. Lest the readers think that these barriers may disappear with advancing technology, we expect the users to drive Moore's law curve down towards even-cheaper systems. However, whether a node produces a wrong reading either after being compromised or due to system failure, both are equally detrimental to the functioning of the network. Security mechanisms developed using cryptographic techniques will not differentiate between readings from faulty and good sensor nodes, resulting into inaccurate results. Cryptography concentrates on providing resiliency against security breaches and such non-malicious activity has to be handled borrowing tools from other domains.

4. REPUTATION BASED FRAMEWORK FOR SENSOR NETWORKS (RFSN)

4.1 Motivation

The problem of generating reliable information in sensor networks can be reduced to one basic question – How do sensor nodes trust each other? We take the motivation from observing the evolution of existing social networks in the world. Embedded in every social network is a web of trust with a link representing the amount of trust between two individuals [24]. Let us analyze the integrated role of "reputation" and "trust" in these networks.

Trust can simply be defined as the expectation of one person about the actions of others. It is used by the first person to make a choice, when an action must be taken before the actions of others are known. *Reputation* is defined as the perception that a person has of another's intentions. When facing uncertainty, individuals tend to trust those which have a reputation for being trustworthy.

RFSN is a similar framework where sensor nodes maintain *reputation* for other nodes in the network. A node monitors the behavior of other nodes, based on which it builds up their reputation over time. It uses this reputation to evaluate their trustworthiness and in predicting their future behavior. At the time of collaboration, a node only cooperates with those nodes that it trusts. The end objective of RFSN is to generate a community of trustworthy sensor nodes.

In a community model, members share some common resources and simultaneously contribute to the community life in order to be entitled to use those resources. In our context, sensor nodes are the members of this community. They contribute to the community life by collaborating in meeting the end-user objective. The network resources which they share are each other. Note that the end-user objective can only be met by collaborative data processing between nodes. A sensor node individually cannot provide any meaningful information to the end user. The key to the development of highly reliable sensor networks lie in making the nodes collaborate with only other good nodes (non-malicious and non-faulty) in the network. Using RFSN, nodes with bad reputation, because they are malicious or are faulty, will be excluded from the community.

It is important to realize that even malicious attacks are carried out by an attacker after seeking the cooperation (unknowingly) of other non-malicious nodes in the network. Let us revisit the network scenario depicted in Figure 1 to verify this assertion. In this scenario, nodes *D*, *E*, *F* or *G* can block packet forwarding only if node *C* chooses the respective compromised node to cooperate by acting as the intermediate relay. Similarly a false negative attack by *A* or *B* is possible only if node *C* takes into account the value reported by them in calculating the final result. Finally, *C* can abuse the system only if *A* and *B* choose node *C* to act as the processing center. A framework based upon reputation and trust will help the nodes to distinguish good nodes from bad, thereby preventing themselves from being exploited by the malicious or failed nodes in the network.

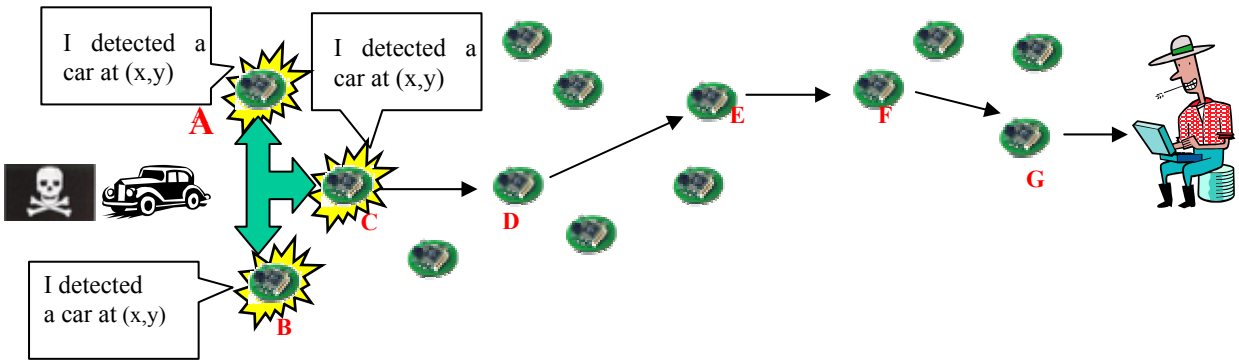

Figure 1: Sensor Network deployed for Intrusion Detection

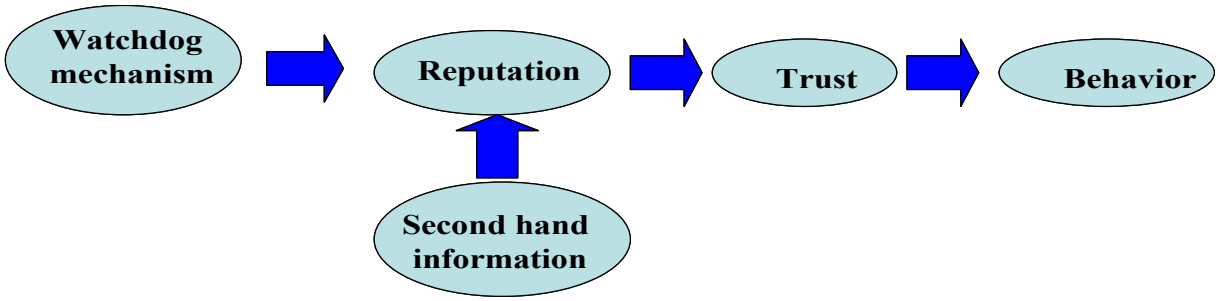

Figure 2: Architectural Design of RFSN

4.2 Architecture

RFSN runs at the middleware of every sensor node and functions in a completely distributed manner. Every node maintains reputation about other nodes (or a subset of them as explained later) and unlike eBay [16], there is no central repository for storing reputation of every entity in the system. Figure 2 depicts the key building blocks of RFSN; the direction of the arrow represents the flow of information between them.

4.2.1 Watchdog Mechanism

The node classifies the actions performed by other nodes in the network as cooperative or uncooperative. This block is responsible for collecting these observations as well as for making the decision, as depicted in Figure 3. This block can be further viewed as a collection of discrete modules. Each module carries out a specific function that can range from monitoring communication channel to sensing channel.

Figure 3 shows an example watchdog mechanism consisting of three modules: *WMRouting*, *WMData* and *WMProcessing*. *WMRouting* monitors the data forwarding behavior of the neighboring nodes by keeping the radio active in the promiscuous mode whereas *WMData* checks for outlier detection by observing the consistency of raw sensing data among the neighboring nodes. Note that each module imposes extra resource requirements on the system in terms of energy, storage or processing cost. The functionality of *WMRouting* not only relies on bi-directionality of communication links but also imposes a system constraint; the radio has to be enabled for promiscuous listening. This is depicted in Figure 3 through a feedback loop connecting the two sub-blocks. Similarly, *WMData* requires knowledge about the spatial-temporal correlation in the underlying physical process which has to be learned by collecting extra data samples over time. Thus, although this block helps a node build reputation over time; it comes at the cost of some resources. Therefore, a judicious design of this block is paramount to the success of RFSN.

4.2.2 Reputation

R_{ij} represents the reputation of node j maintained by node i. Reputation is maintained as a probabilistic distribution, enabling the node to have full freedom and not get constrained by some discrete levels of reputation (+/-1, 0) as used in eBay, Yahoo auctions [25]. Further, maintaining a statistical representation for reputation is more consistent with the model of RFSN. Note that reputation is not a physical quantity but it is a belief; it can only be used to statistically predict the future behavior of other nodes and cannot define deterministically the actual action performed by them.

We define a data structure, termed as the reputation table, RT_i, that stores the reputations maintained by node i.

$$RT_i = \{R_{ij}\} \ldots.(1)$$

A node builds each entry in the reputation table over time through the watchdog mechanism. The interaction between the two blocks is given by equation (2); the output of the watchdog mechanism, D_{ij}, is used to recursively update the reputation of node j at node i, R_{ij}.

$$R_{ij} = f(D_{ij}, R_{ij}) \ldots.(2)$$

We shall quantitatively define D_{ij} anf $f(.)$ in the next section, while describing our system. Qualitatively, D_{ij} represents the rating that is allocated to the latest action of node j by node i and $f(.)$ is responsible for updating the reputation of node j in light of this new observation.

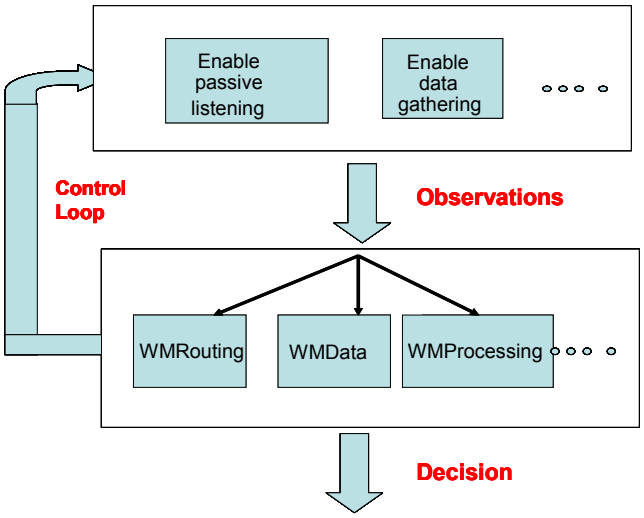

Figure 3: Watchdog Mechanism

4.2.3 Second Hand Information

If a node just relies on its direct observations to build reputation information, the convergence time of the algorithm might be very large. Moreover, these observations will typically impose a huge learning cost. A simple optimization is for nodes to use each others experiences with different nodes in the network. Thus,

nodes exchange reputation information between them and we classify these indirect observations as second hand information.

We explicitly classify the reputation of a node to be a made up of two subcomponents, $(R_{ij})_D$ and $(R_{ij})_{ID}$, as shown in equation (3).

$$R_{ij} = (R_{ij})_D + (R_{ij})_{ID} \quad \ldots(3)$$

We note that summation in equation (3) represent addition of two probabilistic distributions. We name them as direct and indirect reputation respectively. Direct reputation, $(R_{ij})_D$, is build up using direct observations through the watchdog mechanism and indirect reputation, $(R_{ij})_{ID}$, is build up using second hand information respectively. Equation (4) and (5) show the evolution of direct and indirect reputation respectively.

$$(R_{ij})_D = f[D_{ij}, (R_{ij})_D] \forall j \in N_i \quad \ldots(4)$$

$$(R_{ij})_{ID} = (R_{ij})_{ID} + R_{kj} \forall k \in N_i \quad \ldots(5)$$

Although all the direct observations made by a node *i* are equally reliable, the same does not hold for the second hand information. Intuitively, node *i* should give more weight to the second hand information received from a highly reputed node and vice-versa. Thus, the second hand information is appropriately weighed as follows:

$$(R_{ij})_{ID} = (R_{ij})_{ID} + \{w_{ik} * R_{kj}\} \forall k \in N_i \quad \ldots(6)$$

$$w_{ik} = g(R_{ik}) \forall k \in N_i \quad \ldots(7)$$

Here, w_{ik} represents the weight that is derived based on the reputation metric between *i* and *k*, R_{ik}, using *g(.)*.

4.2.4 Trust

Trust is a subjective expectation a node has about another node's future behavior. In RFSN, this is obtained by taking the statistical expectation of the probability distribution representing the reputation between the two nodes. Note that, unlike reputation the trust metric is simply a number.

$$T_{ij} = E[R_{ij}] \quad \ldots(8)$$

4.2.5 Behavior

When faced with the question of cooperating with a node j in the network, the behavior of node *i*, (B_{ij}), is derived from the trust metric between the two nodes. B_{ij} is a binary variable *{cooperate, don't cooperate}* and we use a simple threshold based policy to decide the value of B_{ij}.

$$B_{ij} = \begin{Bmatrix} cooperate \forall T_{ij} \geq TH \\ don't cooperate \forall T_{ij} < TH \end{Bmatrix} \quad \ldots(9)$$

Note that B_{ij} provides a higher-level abstraction; the actual action of node *i* will depend on B_{ij}. For example, reconsider the network scenario in Figure 1, where node *C* has been compromised. If RFSN works perfectly fine, nodes *A* and *B* will calculate B_{AC} and B_{BC} to be *don't cooperate* respectively. They will utilize this information to choose some other data processing center that is trustworthy and not node *C*.

4.3 Localized

Most of the sensor network applications are based on local interactions between nodes that typically lie in the neighborhood of each other. To the best of our knowledge, there exists no sensor network application whereby a node will require prior reputation knowledge about a node many hops distant from it. We note that even if in future some application requires instant reputation information of a distant node, it can be established dynamically at runtime using the chain of trust relationships between neighboring nodes. Thus, from now onwards we will assume that a node maintains reputation information only about its neighboring nodes i.e. nodes that lie in its broadcast domain.

Thereby, propagation of second hand information becomes equivalent to a simple broadcast by a node. The design of the watchdog mechanism also becomes simple; a node needs to monitor just its neighboring nodes. This property of "locality" holds the key for scalability of sensor networks. This same property substantiates our claim of developing reputation based frameworks for highly reliable sensor networks. Not only the nodes need to maintain trust metrics for only a few nodes in the network but they can also easily establish this metric quickly through local interactions.

5. BETA REPUTATION SYSTEM FOR SENSOR NETWORKS (BRSN)

We need mathematical tools to represent reputation, update it continuously based on new direct/indirect observations and finally, make the transition from reputation to the trust metric of a node. In this section, we will lay down the foundations of an example system, based on a Bayesian formulation, being developed for resource constraint sensor nodes within the framework of RFSN.

5.1 Bayesian Formulation

Baye's theorem, equation (10), is used to calculate the probability of a *belief* given an observation.

$$P(Belief / Observation) = \frac{P(Observation / Belief) * P(Belief)}{Normalization} \quad \ldots(10)$$

In a way to make inferences about the unknown quantity, we stipulate a joint distribution *P(observation, belief)*[2] that describes how well these quantities support each other.

Analogously, in RFSN, belief represents the *reputation* of a node and the observation refers to the *direct observations* made by a node about the other node. When node *i* gets some output from the watchdog mechanism, D_{ij} it updates the reputation of node *j*, R_{ij} as follows:

$$R_{ij} = \frac{P(D_{ij} / R_{ij}) * R_{ij}}{\sum P(D_{ij} / R_{ij}) * R_{ij}} \quad \ldots(11)$$

5.1.1 Beta Distribution

Several distributions such as beta, Gaussian, Poisson, binomial etc. can be used to represent the reputation of a node. Among these, the beta distribution has been the most promising due to its

[2] We will use P(.) to refer to the discrete probability as well as continuous probabilistic distribution unless an ambiguity exists.

flexibility and simplicity as well as its strong foundations on the theory of statistics. Recently [26] provided a detailed analysis about reputation based systems based on beta distributions, providing us a good analytical handle over these systems.

The beta distribution is indexed by two parameters *(α, β)*. It can be expressed using the gamma function as:

$$P(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \forall 0 \le x \le 1, \alpha \ge 0, \beta \ge 0 \quad \ldots(12)$$

To analyze the applicability of beta distributions to RFSN, let us impose the constraint that a node rates the behavior of a node during every transaction on a binary scale (cooperative (1), non-cooperative (0)). In general, a node's perceived satisfaction after a transaction does not have to be necessarily binary and we shall generalize our system in later sections.

The basic functionality of RFSN is to predict the future behavior of the nodes. Let us assume that node *i* had interacted with node *j* in *m+n* events; out of which it characterizes *m* and *n* interactions to be cooperative and non-cooperative respectively. Given this information, node *i* want to predict the behavior of node *j* (cooperative/non-cooperative), *θ*, for the next event. Clearly, without any prior information, *θ* is uniformly distributed over the measurement space, (0,1). Thus, P(θ)=uni(0,1)=Beta(1,1). Using the binary rating model, we can model the prior interactions using a binomial distribution and then the posterior distribution of *θ* can be calculated as[3]:

$$P(\theta) = \frac{Bin(m+n, m) * Beta(1,1)}{Normalization} = Beta(m+1, n+1) \ldots(13)$$

Equation (13) shows that the posterior distribution of *θ* is a beta distribution which justifies our choice.

5.1.2 Modeling Reputation & Trust

The reputation of node *j* maintained at node *i* is given by:

$$R_{ij} = Beta(\alpha_j + 1, \beta_j + 1) \quad \ldots(14)$$

Here α_j and β_j represents the cooperative and non-cooperative interactions between node *i* and *j* respectively (from the perspective of node *i*). Without any prior observations, $\alpha_j = \beta_j = 0$ and hence, $R_{ij} = Beta(1,1) = uni(0,1)$. This provides a simple sanity check; without any prior knowledge, the most reasonable reputation function is indeed the uniform distribution. As the beta distribution is defined even over non-integers, equation (14) is valid for all non-negative real numbers, α_j and β_j. We note that equation (13) was derived using a binary model for α_j and β_j in the previous subsection; it is a reasonable assumption for first order analysis.

Thus, reputation table, RT_i at node *i*, contains tuple of the form (α_j, β_j), one corresponding to every node *j* for which node *i* maintains a reputation metric. Note that only two parameters (real numbers) are stored to represent the reputation of a node.

The trust metric of a node is the statistical expectation of the reputation function and is given by:

$$T_{ij} = E(R_{ij}) = E(Beta\{\alpha_j + 1, \beta_j + 1\}) = \frac{\alpha_j + 1}{\alpha_j + \beta_j + 2} \ldots(15)$$

5.1.3 Updating Reputation

The beta function is the conjugate prior for the binomial likelihood distribution. This implies that if the prior distribution is the beta distribution and the new observations follow a binomial distribution, than the posterior distribution will also be a beta distribution. We use this typical property of beta distribution to derive a closed form expression for reputation updates when a node makes some direct observations through the watchdog mechanism.

Let us assume that node *i* has build up some reputation metric, R_{ij}, for node *j*. Node *i* again interacts with node *j* for *r+s* more events; *r* cooperative and *s* non-cooperative. The objective is to find the new reputation of node *j*, R_{ij}. Note that the problem formulation is analogous to that of equation (13), although now the prior distribution is represented by a beta and not a uniform distribution. To achieve a closed form expression let us again constraint *r* and *s* to be integers. The reputation can be updated as:

$$R_{ij} = \frac{Bin(r+s, r) * Beta(\alpha_j + 1, \beta_j + 1)}{Normalization} \quad \text{or}$$

$$R_{ij} = Beta(\alpha_j + r + 1, \beta_j + s + 1) \ldots(16)$$

This clearly shows the flexibility associated with using a beta reputation system. The reputation update is equivalent to just updating the value of the two parameters α_j and β_j as follows:

$$\alpha_j^{new} = \alpha_j + r; \beta_j^{new} = \beta_j + s \ldots(17)$$

We can again relax the constraint of *r* and *s* to be integers and can still approximate the reputation update step by equation (17) for a first order analysis. Note that the output of the watchdog mechanism, D_{ij} should be mapped to a tuple of the form *(r, s)*. However, this mapping can be application specific. For example a node might assign a rating of *(r=1000, s=0)* if node *j* coordinates on an event *A*, significantly affecting the reputation of node *j*. On the other hand, a lower value *(r=1, s=0)* can be assigned for cooperation on event *B*, thereby downplaying the significance of cooperative behavior by node *j*. Similar levels of differentiation can be used to mark the uncooperative behavior of node *j*. Notice that since this is a local procedure; a node can rank the behavior of different nodes based on its own system state without being consistent with other nodes in the system.

5.1.4 Aging

It is intuitive to imagine that the recently obtained information should be given more weight. This is achieved by incorporating exponential averaging, as proposed in [26], in the following way:

$$\alpha_j^{new} = (w_{age} * \alpha_j) + r; \beta_j^{new} = (w_{age} * \beta_j) + s \ldots(17)$$

Here, w_{age}, termed as aging weight, can take values in the range *(0, 1)*. The aging weight is also responsible for making sure that all the nodes cooperate at all the time. A malicious node can very well choose a strategy of cooperating at the start and then abusing the system thereafter using the reputation that it has acquired initially. An appropriate choice of the aging weight will make sure that reputation information becomes stale; a node needs to cooperate continuously to maintain a good reputation.

[3] Due to space constraints we don't provide any statistics proofs. Interested readers can refer to [29].

5.1.5 Second Hand Information

Node i receives reputation information about node j through node k. Let us represent these indirect observations by (α_j^k, β_j^k). Node i already have prior reputation information about j and k, represented by (α_j, β_j) and (α_k, β_k) respectively. We need to combine these pieces of information to obtain new reputation information of j, $(\alpha_j^{new}, \beta_j^{new})$.

The solution to this was given recently in [26]. The approach is to map the problem into an equivalent problem in the realm of Dempster-Shafer belief theory [27] and then solve it using the concept of belief discounting [28]. The final solution is obtained by doing a reverse mapping from belief theory to continuous probability distributions. The closed form expressions, as derived in [26], are given by the following equations:

$$\alpha_j^{new} = \alpha_j + \frac{\{2 * \alpha_k * \alpha_j^k\}}{\{(\beta_k + 2) * (\alpha_j^k + \beta_j^k + 2)\} + \{2 * \alpha_k\}}$$

$$\beta_j^{new} = \beta_j + \frac{\{2 * \alpha_k * \beta_j^k\}}{\{(\beta_k + 2) * (\alpha_j^k + \beta_j^k + 2)\} + \{2 * \alpha_k\}} \quad \ldots(18)$$

A simply sanity check can be done taking the special case of untrustworthy node $(\alpha_k=0)$. As can be observed from equation (18), node i will completely disregard any information from node k in this scenario. Moreover, it can also be verified that a more trustworthy node (higher α or lower β) gets a higher weight in equation (18).

5.2 Good Reputation System

The extra advantage of gaining indirect reputation comes at the cost of making the system vulnerable to *bad-mouthing* attacks. Malicious nodes can deliberately propagate unfair negative ratings about other good nodes, thereby lowering their reputation in the perception of other nodes in the network. This is analogous to a denial of service attack. We remove this attack by allowing the nodes to only propagate good reputation information about other nodes. This resiliency comes at the cost of system efficiency as now nodes cannot exchange their bad experiences about malicious/faulty nodes in the network.

Architecturally, every node divides the reputation table internally into two sub-tables: containing the reputation information about cooperating (RT_i^c) and non-cooperative nodes (RT_i^{nc}) respectively. A node should then propagate only RT_i^c as second hand information. For doing this classification, a node uses the trust and behavior blocks as mentioned in the earlier subsections. The threshold chosen, in equation (9), for this classification is a global parameter, TH_{SHI}. Choosing a globally consistent parameter provides a simple mechanism for crosschecking the validity of second hand information. A node can simply neglect the second hand information with a lower trust value than TH_{SHI}.

5.3 Fresh Information

The reputation table consists of information that has been build up by using both direct as well as indirect observations. Thus, the entries in the reputation table of a node, RT_i^c, are dependent on the entries in other nodes as well. Therefore, if node i propagate RT_i^c as it is, it might result in the same information looping back to generator node. Ideally, we would like to propagate only independent reputation information. In fact, the proof of equation (18) relies on the independence of (α_j^k, β_j^k) and (α_j, β_j).

This can be achieved by propagating only direct reputation $(R_{ij})_D$ as it is generated using only direct observations and is independent of the reputation information received from other nodes. Thereby, we incorporate provisions to exclusively maintain direct reputation, $(R_{ij})_D$, in addition to complete reputation, R_{ij} in the system.

$(R_{ij})_D$ is also maintained as a tuple $\{\alpha D_j, \beta D_j\}$; albeit this tuple exclusively contains the information gained from the direct observations through the watchdog mechanism. Thus, when direct observations are taken by a node i about some other node j, both $\{\alpha_j, \beta_j\}$ (representing R_{ij}) and $\{\alpha D_j, \beta D_j\}$ (representing $(R_{ij})_D$) are updated accordingly. However on getting second hand information about node j, only $\{\alpha_j, \beta_j\}$ (representing R_{ij}) is updated. The data structure that stores $(R_{ij})_D$ is represented by RTD_i, which is also internally divided into two subparts: RTD_i^c and RTD_i^{nc}, using the same global parameter, TH_{SHI}. Node i propagates RTD_i^c instead of RT_i^c as second hand information. Figure 4 shows the resulting architectural design of the Beta Reputation-system for Sensor Networks (BRSN).

Note that unlike RT_i, RTD_i is refreshed $(\alpha D_j = \beta D_j = 0)$ periodically, after a packet broadcast containing RTD_i^c, so that only fresh reputation information gets propagated every time. We note that an ambiguous scenario can exist whereby there is a conflict about the status of a node between RTD_i and RT_i. For example, consider a scenario where node j at a given instant of time lies simultaneously in RTD_i^c and RT_i^{nc}. This scenario is indeed feasible. It physically implies that a malicious node has recently carried out enough cooperative behavior so that it gets characterized as a cooperative node $(j \in RTD_i^c)$. However, this behavior is still not enough to counter its malicious behavior in the past and its total reputation still falls below the threshold $(j \in RT_i^{nc})$. Consistent with our general approach of trading off efficiency with security, we take a pessimistic approach of propagating reputation information only about those nodes that simultaneously lie in both RTD_i^c and RT_i^c.

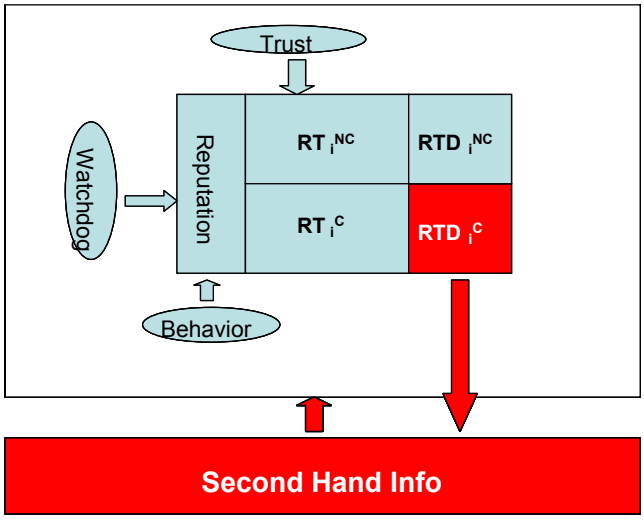

Figure 4: Middleware Design of BRSN

5.4 Node Authentication

In BRSN, the reputation of nodes is associated with their unique identity. Furthermore, the functioning of the watchdog mechanism, reputation updates, propagation of second hand information implicitly assumes the presence of explicit node authentication techniques. We hope to use available prototype implementations, based on cryptography, such as μ-TESLA [4] for authenticated broadcasts and PAKE [8] for pair-wise authentication between neighboring nodes. This also emphasizes the point that RFSN does not eradicate the need of cryptography; both are required for achieving trustworthy sensor networks.

6. ANALYSIS

Reputation based systems have been studied in many contexts and equally diverse domains such as human social networks, e-commerce (eBay [16]), 802.11 networks (confidant [17] and core [18]), peer-to-peer networks ([19]) etc. In this section we contrast BRSN to these existing systems. Besides presenting a qualitative reasoning, we provide a quantitative comparison with a hypothetical system, DUMB-BRSN that blindly combines the second hand information received from other nodes without weighing them appropriately.

6.1 Simulation Setup

We have implemented BRSN and DUMB-BRSN in NESLsim [30], a PARSEC based simulation platform for sensor networks. We consider a network scenario where sensor nodes are scattered randomly to monitor the temperature of a terrain. The data flows upwards from the node towards the base station along the branches of a hierarchical tree structure. We consider a parent-child pair, node j and i respectively, shown in Figure 5(a). We use the trust between nodes i and j, Tij, as a metric for analyzing the efficacy of BRSN and DUMB-BRSN.

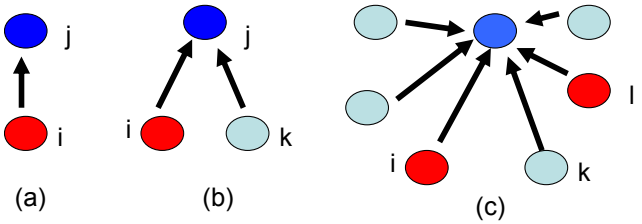

Figure 5: Network Setup

After taking a data sample, node *i* sends this data to node *j* which forwards it towards the base station. We implement a module in the watchdog mechanism, termed as *WMRouting*, whereby node *i* monitor this data forwarding by node *j*. Node *i* initializes the two reputation parameters of node *j*, α_j and β_j, to 0, thereby the initial trust metric, T_{ij}, is 0.5. For each data packet that is successfully forwarded by node *j*, node *i* assigns a positive rating of (1,0) and assigns a rating of (0,1) otherwise. We simulate a perfect channel i.e. no packet is dropped due to network ambiguities. The aging weight, w_{age}, is randomly chosen to be *0.98*.

Let us consider two different scenarios – (1) Node *j* fully cooperates and forwards every packet of node *i* (2) Node *j* is malicious and drops every packet of node *i*. Figure 6 shows the anticipated evolution of trust, T_{ij}, in both the scenarios for BRSN.

We extend the network set up to be consisting of three nodes – *i*, *k* and *j* in the neighborhood of each other, as shown in Figure 5(b). Both nodes *i* and *k* forward their data packet to *j* and monitor the data forwarding by *j*. We currently assume that both *i* and *k* measure the same physical process value. Based on this, we implement *WMEqData* module in the watchdog mechanism. When node *i* overhears the packet from *k*, it checks the data value contained in the packet. If the value is equal to the sensed value of node *i*, it marks this action by node *k* as cooperative (rating of (1,0)) and non-cooperative (rating of (0,1)) otherwise. The threshold (TH_{SHI}) for classifying nodes as good and bad is randomly fixed to be *0.9*.

We again simulate Scenario 1, where node *j* fully cooperates and forwards every packet of nodes *i* and *k*. As can be observed from Figure 6, node *i* make use of the second hand information received from node *k* to establish a higher trust metric with *j*. Note that until the trust metric crosses the threshold of *0.9*, no reputation propagation will take place. This is implicitly shown in Figure 6 as the two curves are aligned with each other until the T_{ij} reaches *0.9*. The impact of second hand information might look small from Figure 6 but this is an artifact of choosing a high threshold value and only one helping neighboring node. We also repeated the experiment for Scenario 2, where node *j* is malicious, and verified that no reputation propagation takes place.

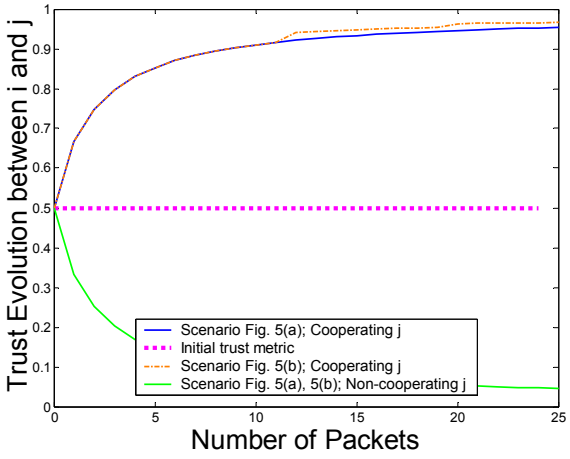

Figure 6: Evolution of trust for BRSN

6.2 Bad Mouthing Attacks

E-commerce system such as eBay, Yahoo auctions etc. have been found to be highly vulnerable to "bad-mouthing attacks", whereby sellers can collude with buyers to drive other sellers out of the market [22]. The conspiring buyers provide unfairly negative ratings to the targeted sellers, thereby lowering their reputation. Reputation-based systems developed for ad-hoc networks such as confidant [17] try to limit the damage of these attacks by relying on trusting nodes to report bad behavior. It uses special messages called *alarm* messages to handle this case. However, this makes the system vulnerable to retaliation attacks, as analyzed in [31]. In BRSN, we follow an approach similar to core [18] and remove this attack completely by allowing the nodes to only propagate good reputation information about other nodes in the network.

6.2.1 Simulation Study

Reconsider the network set up of Figure 5(b); albeit now node *k* has been compromised. Not only node *k* generates no packet for *j* but it also tries to bad mouth *j* by propagating false low reputation information about it. The intention of malicious node *k* is to make node *i* establish a low value for T_{ij} and thereby make node *i* wrongly conclude that node *j* is malicious.

Notice that in absence of any packet from *k*, T_{ik} will remain equal to its initialization value of *0.5*. We always choose the threshold (*0.9* in this case) higher than the initial trust metric of 0.5; node *k* will be automatically classified as malicious by node *i*.

As can be seen from Figure 7, functionality of DUMB-BRSN is highly susceptible to these attacks. Node *i* falsely establishes a low trust value for node *j*. BRSN is resilient to any bad mouthing attacks in the network. It neglects the bad reputation information propagated by node *k*. Node *i* is able to establish exactly the same trust metric as in the scenario when node *k* is absent (Figure 6). BRSN also facilitate automatic evasive action against compromised nodes carrying out bad mouthing attacks in the network. When node *i* get bad reputation information from node *k*, it not only neglects this information but also marks this as a malicious action by node *k*. Note that propagating bad reputation information is inconsistent with the functioning of BRSN and has to be a result of malicious behavior.

To analyze the impact of multiple malicious nodes, we added four more nodes to the network, as shown in Figure 5 (c). We consider the scenario where out of these six children of node *j*, only *i* & *l* are good; rest four including *k* have been compromised. Figure 7 shows the trust evolution for this scenario (Scenario 2). The performance of DUMB-BRSN degrades even more in the presence of additional malicious nodes. BRSN is not only able to maintain its resiliency against bad mouthing attacks by a population of malicious nodes but is also able to selectively take the advantage of one good node.

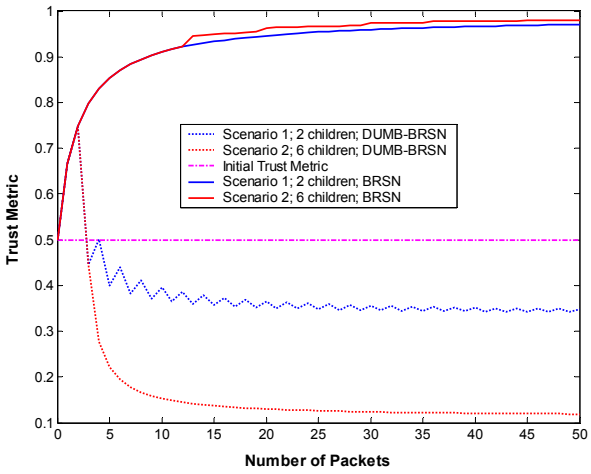

Figure 7: Resiliency to Bad-mouthing Attacks

6.3 Ballot Stuffing Attacks

In an e-commerce system sellers can also collude with buyers in order to obtain unfairly high ratings from them [22]. This will have the effect of inflating a seller's reputation, thereby allowing the seller to receive more orders. This kind of attack can also be achieved in distributed reputation based systems such as core and confidant [15]. BRSN counters this attack by explicitly including the reputation of the reporter node in the integrating step (equation 18).

6.3.1 Simulation Study

We reconsider the network set up in Figure 5(b) but now consider the scenario when both node *j* and *k* have been compromised. Not only node *j* drops every packet from node *i* but it also colludes with node *k* to obtain unfairly high ratings from it. As can be seen from Figure 8, in DUMB-BRSN node *i* establish a good trust metric about node *j* in spite of its non-cooperation. Thereby, node *i* wrongly concludes that node *j* is trustworthy. Contrary to this, BRSN is resilient to any ballot stuffing by node *k*. This is because *k* is malicious from the perspective of *i* and hence, it neglects any reputation information received from *k* although the nature of reputation information is good.

As in the previous section we repeat the experiment for network scenario shown in Figure 5(c). As anticipated, the performance of DUMB-BRSN degrades even more, whereas the performance of BRSN remains unaltered with multiple malicious nodes. Note that unlike the previous case node *i* is unable to take any additional advantage from the presence of one good node, *l*, in the neighborhood. This is the artifact of propagating only good reputation information in the system.

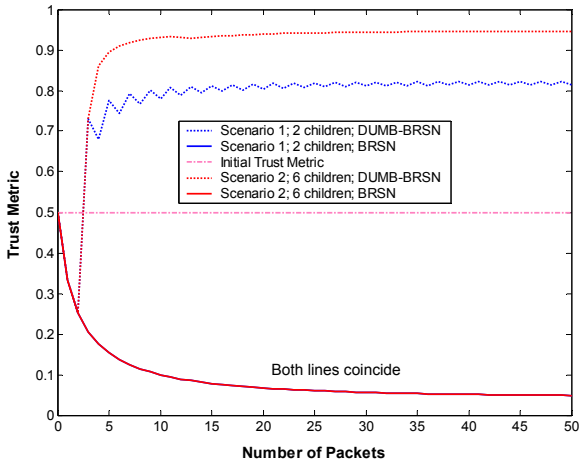

Figure 8: Resiliency to Ballot Stuffing Attacks

6.4 Identity Attacks

Reputation based systems for ad-hoc networks or e-commerce systems are highly susceptible to impersonation attacks, whereby a malicious node tries to impersonate the identity of other good node in the network. The intention of the attacker is to prevent from getting its identity revealed in the network. However, more drastic is the active nature of this attack whereby the intention of the attacker is that the good node should become the victim of any evasive action carried out by other nodes in the network. This is analogous to a denial of service attack.

Unlike eBay or Yahoo auctions, where reputation information for any node is accessible to any other node from the central trust authority, BRSN keeps this information local and secretive to a

node. Nodes do exchange good reputation information about other nodes but unlike confidant, there is no explicit information exchange about malicious nodes in the system. This implies that it is hard for an adversary to realize whether it has been classified as malicious or not, thereby preventing any retaliation strategies such as choosing other nodes, changing identities etc.

Let us still assume that a malicious node somehow learns that it has been classified as a bad node and hence, it adopts a new identity to carry out fresh attacks in the system. An e-commerce system such as eBay or Yahoo auction will completely breakdown in front of these attacks. Both confidant and core do not mention anything explicitly to counter these attacks but their rough proposal is to pre-assign a trust relationship between subset of nodes so that no new identities can be added in the future. Clearly, this approach requires the designer to be aware of the deployed network topology and moreover, does not allow the addition of nodes at run time, making it highly restrictive.

In BRSN, we take an approach similar to the "pay your dues" approach in a community model, whereby an entity has to first perform positive work to gain a good reputation and henceforth it can use the services of other entities in the network. BRSN operates by initializing the reputation to a null value and gradually builds it over time. This clearly undermines the effect of an adversary changing its identity or creating many virtual nodes. All these new nodes (physical/virtual) will be initialized with null reputation to start with and hence, will automatically be characterized as untrustworthy.

This approach works well for BRSN because of the relative static nature of sensor networks. The subset of nodes with which a node interacts seldom changes over time. Moreover, this subset of nodes lies in the neighborhood of a node so that reputation can be build up only through local interactions. Furthermore, sensor networks are envisioned to be highly dense so that this subset consists of a significant number of nodes. Thus, using BRSN sensor nodes are quickly able to establish meaningful reputation information about each other and hence, seldom face the case of cooperating with a complete stranger, except during the small initialization phase.

Unfortunately, a similar approach is not feasible in e-commerce systems or traditional ad-hoc networks where cooperation might have to take place between complete strangers. Moreover, the mobility of nodes can be significant in ad-hoc networks so that the subset of nodes with which a node needs to interact continuously changes over time. Thus, a node would again need to somehow generate instant reputation information about a distant entity without having any past interaction with it.

7. FEATURES & FUTURE CHALLENGES

In this section, we analyze some of the features of RFSN and make a compelling case to develop systems such as BRSN for realizing trustworthy sensor networks. We then outline some challenges for the realization of RFSN and introduce some promising approaches that we are currently pursuing.

7.1 Features

7.1.1 Generalized

Several customized solutions, based on cryptography, for providing secure communication, aggregation, data integrity, access control etc. have been developed in the realm of sensor networks. All these solutions can be incorporated in the watchdog mechanism of RFSN as discrete modules. RFSN can integrate all of them to provide one classifying metric, reputation, for a node. Cryptographic schemes such as SERP, SIA, SPINS etc. can then exploit this reputation information, build using RFSN, to decide the subset of nodes with which to interact while doing key establishment, generating secure event reports etc. in future. Thus, RFSN and these customized solutions can work in conjunction with each other to provide a complete solution for high integrity sensor networks.

7.1.2 Unified

In RFSN, a sensor node act upon any inconsistent behavior without caring about the origin of it. From a network perspective, both malicious and faulty behavior is equally detrimental and hence, should be acted upon in a similar fashion. Temporary system faults such as interim malfunctioning of sensors or radio and temporary network faults such as fading on the communication channel will span over a small duration of time and maintaining a probabilistic distribution for reputation will automatically filter out such sporadic behavior. Permanent faults will indeed be dealt in the same manner as malicious nodes.

7.1.3 Diversity

Different applications can be provided with varied security options by setting application specific threshold values for judging the trustworthiness of a node. For example an application that finds the maximum temperature value in a room can have relatively smaller threshold value compared to an intrusion detection application. Furthermore, the requirements of the security level can be changed dynamically, say after detecting a breach, by runtime update of the threshold. The design of RFSN does not impose any constraints on the threshold value and hence it can be updated accordingly by the network designer.

7.1.4 Scalability

The static and the localized nature of sensor networks provide scalability to RFSN. Not only it is sufficient for nodes to maintain reputation for only a few nodes but they can establish these metrics quickly and easily through local interactions.

7.2 Challenges

7.2.1 Watchdog Mechanism

This block helps a node build reputation over time; albeit at the cost of some resources. Therefore, a judicious choice of modules is paramount to the success of RFSN. We are currently investigating existing and novel challenge-response protocols, outlier-detections schemes and data analysis protocols to develop these modules. We envision running this block in a customized manner; modules are available as APIs and it is the responsibility of the end-user to enable a subset of them. Moreover, the system design should allow an easy runtime insertion/removal of these modules.

7.2.2 Bootstrapping

RFSN takes a pessimistic approach at the onset of the network, whereby no node in the network trusts each other. The reputation gradually builds up over time. An inherent assumption made is that there exists significant opportunities in the network whereby

nodes can learn about each other. However, there exist sense-response applications where the expected network activity is low. Thus, a mechanism is needed to pro-actively establish trust among nodes. We are pursuing the direction of using mobile trustworthy nodes as a bootstrapping mechanism for trust establishment. These nodes can be used to fabricate some events in the network, thereby providing opportunities for the nodes to monitor each other's behavior.

7.2.3 Hierarchical Structure

RFSN operates on the basic principle of Bayesian decision theory; past behavior of a node can be used to predict its future behavior. This can be exploited by intelligent context aware adversaries. They can potentially compromise a highly reputed node at the runtime and then can use it to abuse the system. Similarly, the development of several watchdog modules is based on the fact that majority of nodes in the neighborhood have not been compromised.

In a nutshell, there are limits to which a homogeneous system can provide security; some form of hierarchy is needed. For instance, some high-end trusted nodes can be deployed to periodically check the status of nearby nodes. Similarly a secure data mule can be made to periodically traverse through the network to perform this status check. This opens a lot of new issues – What should be the density of these high-end nodes? What should be the period of data mule? We intend to do an algorithmic as well as statistical study to address these challenges.

7.2.4 Good Reputation System

RFSN has been designed as a good reputation system; disseminating bad reputation information is prohibited. Although this approach counters the bad-mouthing attacks affectively, it comes at the cost of system efficiency as nodes cannot share their bad experiences with each other. In [21], a promising approach of maintaining reputation of reputation ratings has been proposed. The basic concept is to take into account the consistency of the latest received reputation information with the past information during the integration step besides the nature (good/bad) of it. Authors in [21] just provide a very high level framework for this based on trivial heuristics. Future work involves developing a concrete statistical foundation for this approach. We believe that if fully developed this will allow disseminating both good and bad reputation information in the network without making the system vulnerable to bad-mouthing or ballot stuffing attacks.

8. CONCLUSIONS

Cryptography presents an efficient mechanism for node authentication and maintaining data confidentiality and integrity. We highlight some novel characteristics of these networks leading to unconventional attacks and system failures where cryptographic solutions are not sufficient. For example, cryptography cannot provide data authentication needed for countering misbehavior from internal adversaries and faulty nodes. On the basis of these observations, we motivate the need of integrating tools from different domains such as economics, statistics and data analysis with cryptography to facilitate the development of high integrity sensor networks.

Following this approach, we propose a framework, RFSN, for developing a community of trustworthy sensor nodes at runtime based upon the behavior of these nodes. Sensor nodes maintain reputation for other nodes and use it to evaluate their trustworthiness. RFSN provides a scalable, diverse and a generalized approach for countering all types of misbehavior resulting from malicious and faulty nodes in the system. The lightweight modular architecture of RFSN has been designed keeping into mind the resource constraint nature of sensor nodes.

Within the framework of RFSN, we have developed a beta reputation system for sensor networks (BRSN) that uses a Bayesian formulation for reputation representation, updates, integration and trust evolution. We have provided a detailed analysis of our system design choices, contrasting them with existing reputation-based systems from e-commerce and ad-hoc networks. The efficacy of BRSN is verified through some preliminary simulation results. Based on these results we claim that RFSN provides a practical solution for developing high integrity sensor networking systems.

9. ACKNOWLEDGMENTS

This material is based on research supported in part by the Center for Embedded Networked Sensing (CENS), a NSF Science & Technology Centre, and by the Office of Naval Research (ONR) under the AINS Program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the CENS or the ONR.

10. REFERENCES

[1] C. Karlof, D. Wagner. Secure routing in sensor networks: Attacks and countermeasures. *Elsevier AdHoc Networks journal*, May 2003.

[2] A. Perrig, J. Stankovic, D. Wagner. Security in Wireless Sensor Networks. *Communications of the ACM*, 2004.

[3] J. Newsome, E. Shi, D. Song, A. Perrig. The sybil attack in sensor networks: Analysis and Defenses. *In Proceedings of IPTPS.* March 2002.

[4] A. Perrig, R. Szewczyk, V. Wen, D. Culler, D. Tygar. SPINS: Security Protocols for Sensor Networks. *Wireless Networks Journal*, September 2002.

[5] C. Karlof, N. Sastry, D. Wagner. TinySec: Link Layer Encryption for Tiny Devices. *To appear in ACM SenSys*, 2004.

[6] J. Deng, R. Han and S. Mishra. The Performance Evaluation of Intrusion-Tolerant Routing in Wireless Sensor Networks. *In the Proceedings of IPSN*, April, 2003.

[7] R. Watro, D. Kong, S. F. Cuti, C. Gardiner, C. Lynn, P. Kruus. TinyPK: Securing Sensor Networks with Public Key Technology. *To appear in second workshop on Security in Sensor and Ad-hoc Networks*, 2004.

[8] S. Ganeriwal, R. Kumar, C. C. Han. S. Lee, M. B. Srivastava. Location & Identity based Secure Event Report Generation for Sensor Networks. *NESL Techanical Report*, May 2004.

[9] F. Ye, H. Luo, S. Lu, L. Zhang. Statistical En-route Detection and Filtering of Injected False Data in Sensor Networks. *In Proceedings of IEEE Infocom*, 2004.

[10] L. Eschenauer, V. D. Gligor. A key Management Scheme for Distributed Sensor networks. *In Proceedings of ACM CCS*, November 2002.

[11] H. Chan, A. Perrig, D. Song. Random Key Predistribution Schemes for Sensor Networks. *In Proceedings of IEEE Symposium on Security and Privacy*, 2003.

[12] D. Liu, P. Ning. Establishing pairwise keys in distributed sensor networks. *In Proceedings of ACM CCS*, October 2003.

[13] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized Trust Management. *In Proceedings of IEEE Conf. Security and Privacy*, 1996, Oakland, California, USA.

[14] M. Blaze, J. Feigenbaum, J. Ioannidis and A. Keromytis. RFC2704 - The KeyNote Trust Management System Version 2. 1999.

[15] N. Li, J. Mitchell, and W. Winsborough. Design of a role-based trust management framework. *In Proceedings of the IEEE Symposium on Security and Privacy*, Oakland.

[16] P. Resnick, R. Zeckhauser. Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. *NBER workshop on empirical studies of electronic commerce*, 2000.

[17] S. Buchegger, J. L. Boudec. Performance analysis of the CONFIDANT protocol. *In Proceedings of ACM Mobihoc*, 2002.

[18] P. Michiardi, R. Molva. CORE: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks. *Communication and Multimedia Security*, September, 2002.

[19] L. Xiong, L. Liu. A reputation-based trust model for peer-to-peer ecommerce communities. *IEEE conference on e-commerce*, 2003.

[20] S. Buchegger, J. L. Boudec. Coping with false accusations in misbehavior reputation systems for mobile ad-hoc networks. *EPFL technical report*, 2003.

[21] S. Buchegger, J. L. Boudec. A Robust Reputation System for P2P and Mobile Ad-hoc Networks. *In Proceedings of P2PEcon 2004*, Harvard University, Cambridge MA, U.S.A., June 2004.

[22] C. Dellarocas. Mechanisms for coping with unfair ratings and discriminatory behavior in online reputation reporting systems. *In Proceedings of ICIS*, 2000.

[23] B. Przydatesk, D. Song, A. Perrig. SIA: Secure Information Aggregation in Sensor Networks. *In Proceedings of ACM SenSys*, 2003.

[24] R. L. Trivers. The evolution of reciprocal altruism. *Quarterly review of biology*, 46:35-57.

[25] P. Resnick, K. Kuwabara, R. Zeckhauser, E. Friedman. Reputation systems: Facilitating trust in e-commerce systems. *Communications of the ACM*, 43(12): 45-48.

[26] A. Jsang and R. Ismail. The Beta Reputation System. *In Proceedings of the 15th Bled Electronic Commerce Conference*, June 2002.

[27] G. Shafer. A mathematical theory of evidence. *Princeton University,* 1976.

[28] A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, June 2001.

[29] Beta distribution from Mathword. http://mathworld.wolfram.com/BetaDistribution.html

[30] S. Ganeriwal, V. Tsiatsis, C. Schurgers, M. B. Srivastava. NESLsim: A Parsec based Simulation Platform for Sensor Networks. http://www.ee.ucla.edu/~saurabh/NESLsim

[31] F. Stajano and R. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. *In Proceedings of seventh International Security Protocols Workshop*, 1999.