Cluster-based Reputation and Trust for Wireless Sensor Networks

Garth V. Crosby and Niki Pissinou
Florida International University
Electrical & Computer Engineering Department,
Telecommunication and Information Technology Institute,
Miami, Florida
garth.crosby1@fiu.edu, pissinou@fiu.edu

***Abstract*- Using a reputation-based trust framework for wireless sensor networks we introduce a mechanism that prevents the election of compromised or malicious nodes as cluster heads, through trust based decision making. We employ a secure cluster formation algorithm to facilitate the establishment of trusted clusters via pre-distributed keys. Reputation and trust is built over time and allow the continuation of trusted cluster heads elections. We performed an evaluation of our approach through simulations. The results indicate clear advantages of our approach in protecting the information of our network by preventing the election of untrustworthy cluster heads.**

Keywords: Reputation, Trust, Cluster, Sensor Networks, Beta Distribution

1. Introduction*

Reputation and Trust is the basis of every interaction that requires the performance of a future task based on past behavior. Trust and reputation have become important topics of research in many fields including psychology, philosophy, economics, and computer science. Expert researchers have employed definition appropriate to their respective field. We rely on the following definitions of these two terms.

Reputation: perception that an agent creates about another agent's intention and norms, through direct or indirect observation of its' past actions [1].

Trust: a subjective expectation an agent has about another's future behavior with respect to a specific action. This expectation can be influenced by many factors including physical characteristics, identity, past behavior and reputation. We focus on behavioral trust evidences and reputation in this work.

Trust should not be reduced to mere security. The latter can be useful to protect from the intrusion of an unknown agent (access control), to guarantee an agent of the identity of its partner (authentication), to identify the sender and receiver of the message (non-repudiation) and to prevent snooping (confidentiality). However, the issue of trust is more complex. Trust must supply us with the necessary tool for making decision, conducting various tasks, and establishing relationships in a world that is intrinsically insecure and with people (entities) whose identity, history or relationship are unknown [2].

Our reputation based trust model is dynamic, that is, trust evidences are constantly assessed and allowed to update a trust metric. Reputation in our work is a probabilistic distribution similar in nature as found in [3, 4]. We employ a data structure that stores the trust values in a trust table maintained by each node. Each node builds and maintains its trust table by monitoring its immediate neighbors.

Clustering provides one of the best solutions for communication in sensor networks due to its inherent energy saving qualities and its suitability for highly scalable networks. Clustering naturally facilitates data aggregation, an energy efficient technique where nodes forwards to a cluster head for processing and fusion before transmitting to base station. Clustering can be extremely effective in multicast, anycast, or broadcast communication. However, to the best of our knowledge, all of the cluster based protocol and cluster formation algorithm that have been proposed assume that the wireless sensor nodes are trustworthy [5, 6]. This assumption may naturally lead to the selection (or election) of a compromised or malicious node to be the cluster head. Having a malicious cluster-head severely compromises the security and usability of the network.

It has been demonstrated [7] that if 5% of the nodes misbehave then more than 60% of the routes in a grid sensor network and more than 35% of the routes in a randomly placed sensor network, would be infected. For 10% of misbehaving nodes the figures are 88% and 54% respectively [7]. These results imply that in a cluster-based protocol such as LEACH in which optimally 5% of the nodes are cluster heads[5], it is likely that a significant portion of the network can be paralyzed or the entire network disabled, in the worst case scenario, if these cluster heads are compromised.

Our main contribution in this paper is our novel approach in maintaining trusted clusters through a trust-based decision making cluster head election algorithm. The remainder of this paper is organized as follows. In section 2, we describe the probabilistic models, which are similar to [4], that we employed. In section 3 we describe our distributed trust framework and cluster head election mechanism. In section 4,

* This work was supported in part by grants from the U.S. Dept of Defense, U.S. Dept. of Transportation and the National Science Foundation.

we present our simulations and analyses. We conclude in section 5.

2. Probabilistic Model

2.1 Notation

In a wireless sensor network consisting of n nodes, we denote the set of all nodes as $S = \{s_1, s_2, ..., s_n\}$. After deployment pairs of nodes $\{s_i, s_j\} \subseteq S$ may interact directly with each other in order to perform a specific task that requires cooperation. Such an interaction may be considered successful by s_i if s_j cooperates in the performance of the task. The history of observed outcome between s_i and s_j, from the perspective s_i, is recorded at any time t as a tuple, $H^t_{s_{ij}} = (c^t_{s_{ij}}, d^t_{s_{ij}})$ where the value of $c^t_{s_{ij}}$ is the number of successful interaction (cooperation) of s_j with s_i, while $d^t_{s_{ij}}$ is the number of unsuccessful interactions.

2.2 Beta Distribution

Various distributions such as beta, binomial, Poisson, Gaussian, etc. have been used to represent the reputation of an agent (node). In recent times, the beta distribution has been employed in a number of works [3, 4, 8] . Jøsang [4], in particular, has provided a thorough treatment of beta distribution and its usefulness in reputation systems. We opted to use beta distribution because of its simplicity, strong foundation on statistical theory, and the fact that its computation requires mainly two shape parameters which make it quite applicable for the memory constrained wireless sensor nodes and, its appropriateness in representing the probability distribution of binary events.

The beta probability density function $f(p \mid v, \omega)$ can be expressed using the gamma function Γ as:

$$f(p \mid v, \omega) = \frac{\Gamma(v+\omega)}{\Gamma(v)\Gamma(\omega)} p^{v-1}(1-p)^{\omega-1}, \quad \text{where}$$

$0 \le p \le 1, v > 0, \omega > 0,$

with the restriction that the probability variable $p \neq 0$ if $v < 1,$ and $p \neq 1$ if $\omega < 1.$

Let us consider the interaction of two nodes s_i and s_j, from the perspective of s_i there are two possible outcomes $O_{s_{ij}} = 1$ for successful interaction and $O_{s_{ij}} = 0$ for unsuccessful interaction. In this context $c^t_{s_{ij}}$ and $d^t_{s_{ij}}$, which were defined previously also mean that the outcome $O_{s_{ij}}$=1 was observed $c^t_{s_{ij}}$ times and $\overline{O_{s_{ij}}}$ was observed to occur $d^t_{s_{ij}}$ times. The probability density function of observing outcome $O_{s_{ij}}$=1 in the future can be expressed as a function of past observations by setting:

$v = c^t_{s_{ij}} + 1$ and $\omega = d^t_{s_{ij}} + 1$, where $c^t_{s_{ij}}, d^t_{s_{ij}} \ge 0$.

The expectation value for the beta distribution is defined as: $E(p) = v/(v+\omega)$, where p is probability variable.

2.3 Modeling Reputation

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

$R^t_{s_{ij}} = \frac{\Gamma(v+\omega)}{\Gamma(v)\Gamma(\omega)} p^v(1-p)^\omega$, where $0 \le p \le 1, v > 0, \omega > 0$;

setting $v = c^t_{s_{ij}} + 1$ and

$\omega = d^t_{s_{ij}} + 1$, where $c^t_{s_{ij}}, d^t_{s_{ij}} \ge 0$.

2.4 Modeling Trust

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

$$T_{s_{ij}} = E(R^t_{s_{ij}}) = \left(c^t_{s_{ij}} + 1\right) / \left(c^t_{s_{ij}} + d^t_{s_{ij}} + 2\right)$$

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

2.5 Updating Reputation

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

$$c^{t+q}_{s_{ij}} = c^t_{s_{ij}} + c^{(t+q)-t}_{s_{ij}} \; ; \; d^{t+q}_{s_{ij}} = d^t_{s_{ij}} + d^{(t+q)-t}_{s_{ij}}$$

$$R^{(t+q)}_{s_{ij}} = \text{Beta}(c^{t+q}_{s_{ij}} + 1, d^{t+q}_{s_{ij}} + 1)$$

3. Distributive Trust-based Framework

Our primary goal is to develop a reputation based trust framework for cluster-based wireless sensor networks and, a mechanism that reduces the likelihood of compromised or malicious nodes being selected (or elected) as cluster heads. We make a number of assumptions. Firstly, a reliable link

layer protocol and cluster formation algorithm is assumed. Once the clusters are formed they maintain the same members, except for cases where nodes are blacklisted, die or when new nodes join the network. All the nodes communicate via a shared bidirectional wireless channel and operate in the promiscuous mode. We do not consider key distribution but we assume that each node has three keys; a master, cluster and pairwise. The master key is shared by every node and facilitate broadcast by the base station. Members of each cluster share the cluster key. Each cluster has a different cluster key. This key facilitates multicasting communication from the base station to a cluster and also group communication within the clusters themselves. The pairwise key allows node-to-node communication.

3.1 Threat Model

We have considered a motivated attacker that attempts to become a cluster head via malicious or compromised nodes after the setup phase of the network. We envision that non-critical commodity wireless sensor nodes (non-military and non-mission critical applications) will be cheap, under a dollar per node. As such, it would not be cost effective to implement tamper proof techniques in these nodes. As a result of this, it would be quite possible for a motivated attacker to recover valuable cryptographic information through physical extraction and then redeploy these nodes in the network.

3.2 Cluster Head Election Mechanism

In our scheme the cluster head performs the usual functions such as data aggregation, fusion and higher level transmission to the base station. We employ an algorithm similar to the one first proposed by Dimitriou et al [9], to form our initial clusters. (For the details please consult [9]). This algorithm enables the establishment of trusted clusters in the initial stages of the network through the use of pre-distributed key. After the formation of our clusters each node monitors and records the behavior of its immediate neighbors in a trust table.

When the current cluster head's battery power level falls below a predetermined threshold or serve for a predetermined period of time, it broadcasts (within the cluster) a *new election* message. All the nodes then vote for a new cluster head by using secret ballot. This is done by replying to the *new election* message with its choice of candidate. The reply, or vote, is encrypted with the pairwise key with the cluster head. Neighbors therefore have no idea of the political affiliation of each other since the key is private and, different for each node–cluster head pair. The top pick from its list of trusted neighbors is selected as the node's candidate. The current cluster head then tallies the votes and decides the winner based on simple majority. The node with the second highest number of votes is selected as the vice cluster head. The purpose of the vice cluster head is to assume cluster head function in the event that the newly elected cluster head fails before handing over to its successor. At the completion of tallying, the cluster head multicast the winner and runner-up to all the members of the cluster.

For greater integrity the new winner and runner-up have to pass a challenge-response from the cluster head before they are allowed to take up office. To prevent false positives,

typically 2-3 challenges would be issued if there is no timely response. If one or both of them fail the challenge-response the incumbent cluster head informs the cluster members and, initiate a new election for the replacement of the node(s), which did not pass the challenge-response. The failed node(s) are blacklisted in the cluster nodes' and members trust tables by setting its trust level value to -1. Once a node is set to -1 no further trust level update is done and no future interaction takes place with that node.

Periodically, the cluster head will broadcast a *not trusted* message. In this case, nodes select the least trusted neighbor and reply to the cluster head in a similar manner to the voting process. The cluster head tallies the *no trust* messages and selects the node that is least trusted by the most nodes with confidence metric above predetermined value. That node is then given a challenge-response by the cluster head. If it fails, it is blacklisted. If it passes, the cluster members are informed as such. However, they are not obliged to improve the trust level of the node in question because it may not be malicious and or compromised but may still be unreliable and as such deserves a low trust level.

The procedure in Figure 1 gives a high level description of the action of the current cluster head in the election of a new cluster head. A similar procedure applies when electing the vice cluster head.

```
 if power_level() <= threshold or clusterhead_duration >=
predetermined_time
{
New_Election( ) {
        broadcast new_election( )
        count nominees( ) //tally the votes for //each nominee
    if Tie
      top_nominee= randomly_select_nominee( )
    else
        top_nominee= max_count( )
    end if
    //sends challenge response to top_nominee
    if challenge_response( ) =pass
        new_head = top_nominee
        broadcast new_head
    else
        blacklisted=top_nominee
        broadcast blacklisted
        New_Election( )
    end if
end} // end of function New_Election
 }
```

Figure1. Cluster head election procedure

4. Simulation

In this section, we use simulation to study the performance of our model. We use OPNET [10] as our main simulation platform. First, we assessed the capability of our model in preventing compromised nodes from being selected as the cluster head. We then evaluated the power consumption requirement of our model.

4.1 Environment Setup

In our setup, a 20 node cluster is randomly deployed in $50m^2$ area. A free space propagation model is assumed with a data rate set a 2Mb/s. Packet lengths are 10kbit for data packets. The data packets are generated every one second [11]. In addition, we include additional nodes presumably from other nearby clusters. These nodes transmit at 10kbps to a random subset of nodes in the cluster, which are within their transmission range. These additional nodes are presumably for the purposes of relaying data from nearby clusters. We interpret all transmission of these nodes as 'data received for forward'. A node is viewed as cooperative if it relays the 'data received for forward' and uncooperative otherwise. We use a simple TDMA based MAC with only data packets and two types of control packets.

The cluster head runs our cluster election algorithm. We omit the challenge response procedure, assuming that once selected the new cluster head has the necessary cryptographic material. This narrows our study to compromised nodes as oppose to compromised and malicious nodes. We were interested in testing the capability of our algorithm in discerning between trusted and untrustworthy nodes. Therefore, compromised nodes were systematically introduced in the setup by setting the node's packet drop rate to 45%. The packet drop probabilities of the other nodes were set to 0.01. The compromise nodes ignore the prescribed selection routine and randomly votes for nodes. This was implemented since by intuition we do not expect compromised nodes to report truthfully. In the next section, we present results that show the capability of the algorithm in preventing the selection of compromised nodes as cluster heads.

4.2 Analysis of Results

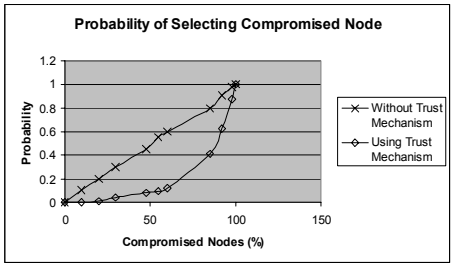

Figure 2. Probability of Selecting Compromised Node as CH

Figure 2 shows the advantage of our selection mechanism over a similar cluster that doesn't employ our trust-based election mechanism. For clusters with less than 17% of compromised nodes our mechanism almost never selects a compromised node. This demonstrates the effectiveness of our mechanism in securing cluster based wireless sensor networks. There is an expected linear increase over time, however, the probability increase exponentially after 60% of the nodes were compromised. This can be explain by an accumulation of errors at the node that makes it increasingly difficult to discern between compromised nodes and uncompromised node in light of the packet drop rate and the false voting of compromised nodes.

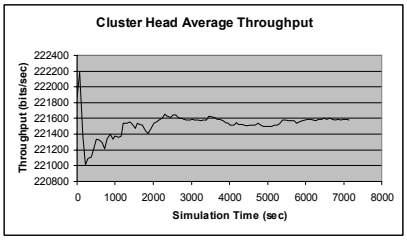

Figure 3. Average Cluster Head Throughput

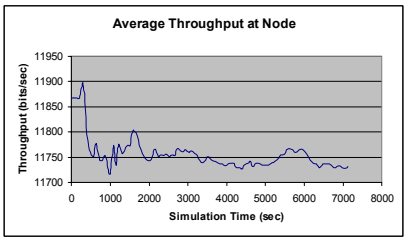

Figure 4. Average Node Throughput

Figures 3 and 4 show the average throughputs of the cluster-head node and a regular node. The average the throughput was approximately 11,750 bits/sec. Based on these results and using the communication energy model in [5] we can obtain some estimate for the power consumption of our model. As an example, if a 1-volt AAA battery with 750mWh is used for each node, the battery can last for 18 days assuming that the node serves a short period as a cluster head. This is a fairly good lifetime for the node given that we have employed a simple MAC, without any energy optimization algorithm.

5. Conclusion

This paper describes a reputation based trust framework with a mechanism for the election of trustworthy cluster heads. Our trust framework is design in the context of a cluster based network model with nodes that have unique local IDs. We assess our model based on power consumption and its ability to prevent compromised nodes from becoming cluster heads. Our approach decreases the likelihood of malicious or compromised nodes from becoming cluster heads.

References

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

[2] R. Falcone, M. Singh, and Y.-H. Tan, "Introduction:Bringing Together Humans and Artificial Agents in Cyber-societies: A New Field of Trust Research," in *Trust in Cybersocieties- Integrating the Human and Artificial Perspectives*, *Lecture Notes in Artificial Intelligence*, R. Falcone, M. Singh, and Y.-H. Tan, Eds. Berlin: Springer-Verlag, 2001, pp. 1-7.

[3] S. Ganeriwal and M. B. Srivastava, "Reputation-based Framework for High Integrity Sensor Networks," presented at ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04), Washington, D.C., USA, October 25, 2004.

[4] A. Jøsang and R. Ismail, "The Beta Reputation System," presented at the 15th Bled Electronic Commerce Conference, Bled, Slovenia, 2002.

[5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishan, "Energy Efficient Communication Protocol for Wireless Microsensor

Networks," presented at The 33rd International Conference on System Sciences (HICSS 2000), Hawaii, 2000.
[6] A. Manjeshwar and D. P. Agrawal, "TEEN: A Protocol for Enhanced Efficiency in Wireless Sensor Networks," presented at The 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, April 2001.
[7] S. Tanachaiwiwat, P. Dave, R. Bhindwale, and A. Helmy, "Location-centric isolation of misbehavior and trust routing in energy-constrained sensor networks," presented at 2004 IEEE International Conference on Performance, Computing, and Communications, 2004.
[8] J. Patel, W. T. L. Teacy, N. R. Jennings, and M. Luck, "A Probabilistic Trust Model for Handling Inaccurate Reputation Sources," presented at the Third International Conference on Trust Management, Rocquencourt, France, 2005.
[9] T. Dimitriou and I. Krontiris, "A Localized, Distributed Protocol for Secure Information Exchange in Sensor Networks," presented at The 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05), Denver, Colorado, 2005.
[10] www.opnet.com.
[11] K. Arisha, M. Youssef, and M. Younis, "Energy-Aware TDMA-Based MAC for Sensor Networks," presented at The IEEE Integrated Management of Power Aware Communications, Computing and Networking (IMPACT'02), New York City, New York, May 2002.