Malicious Node Detection in Wireless Sensor Networks using Weighted Trust Evaluation

Idris M. Atakli, Hongbing Hu, Yu Chen*
SUNY – Binghamton
Binghamton, NY 13902, USA
{iatakli1, hhu1, ychen}@binghamton.edu

Wei-Shinn Ku
Auburn University
Auburn, AL 36849, USA
weishinn@auburn.edu

Zhou Su
Waseda University
Tokyo 169-8555, Japan
zhousu@asagi.waseda.jp

Keywords: Wireless sensor networks, network security, hierarchical topology, malicious node detection.

Abstract

Deployed in a hostile environment, individual nodes of a wireless sensor network (WSN) could be easily compromised by the adversary due to the constraints such as limited battery lifetime, memory space and computing capability. It is critical to detect and isolate the compromised nodes in order to avoid being misled by the falsified information injected by the adversary through compromised nodes. However, it is challenging to secure the flat topology networks efficiently because of the poor scalability and high communication overhead. On top of a hierarchical WSN architecture, in this paper we proposed a novel scheme based on *weighted-trust evaluation* to detect malicious nodes. The hierarchical network can reduce the communication overhead between sensor nodes by utilizing clustered topology. Through intensive simulation, we verified the correctness and efficiency of our detection scheme.

1. INTRODUCTION

Recent advancements in *micro-electro-mechanical systems* (MEMS) and low power and highly integrated electronic devices have led to the development and wide application of *wireless sensor networks* [5], [14], [16]. Wireless sensor networks consist of very small devices, called sensor nodes, that are battery powered and are equipped with integrated sensors, a data-processing unit, a small storage memory, and short-range radio communication [17]. Typically, these sensors are randomly deployed in the field. They form an unattended wireless network, collect data from the field, partially aggregate them, and send them to a sink that is responsible for data fusion. Sensor networks have applications in emergency-response networks, energy management, medical monitoring, logistics and inventory management, and battlefield management.

In contrast to traditional wireless networks, special security and performance issues have to be carefully considered for sensor networks [21]. For example, due to the unattended nature of sensor networks, an attacker could launch various attacks and even compromise sensor devices without being detected. Therefore, a sensor network should be robust against attacks, and if an attack succeeds, its impact should be minimized. In other words, compromising a single sensor node or few sensor nodes should not crash the entire network.

Another concern is about energy efficiency. In a WSN, each sensor node may need to support multiple communication models including unicast, multicast, and broadcast. Therefore, due to the limited battery lifetime, security mechanisms for sensor networks must be energy efficient [19]. Especially, the number of message transmissions and the amount of expensive computation should be as few as possible.

In fact, there are a numbers of attacks an attacker can launch against a wireless sensor network once a certain number of sensor nodes have been compromised. In literature, for instance, HELLO flooding attacks [9], sink hole attacks [9], Sybil attack [12], black hole attack [15], worm hole attacks [6], or DDoS attacks [4] are options for an attacker. These attacks lead to anomalies in network behaviors that are detectable in general. There are some reported solutions to detect these attacks by monitoring the anomalies [9].

In this work, we addressed an even trickier scenario. When an adversary has gained control over certain sensor node(s), he/she does not launch direct attacks against the network. Since once the misbehavior is detected, the operator may forsake these compromised nodes and turn to other data sources. Instead, the attacker let those compromised nodes behave normally but report false data to the data collector. The purpose of the adversary is to mislead the operator with falsified data. This may lead to more serious consequences; for instance, in the battlefield a false report regarding the operations of the enemy may lead to extra casualties.

---

*Manuscript submitted on Jan. 11, 2008 to *The Symposium on Simulation of Systems Security* (SSSS'08), Ottawa, Canada, April 14 –17, 2008. Corresponding author: Yu Chen, Dept. of Electrical & Computer Engineering, SUNY – Binghamton, Binghamton, NY 13902. E-mail: ychen@binghamton.edu, Tel.: (607) 777-6133.

In this paper, we proposed a *weighted-trust evaluation* (WTE) based scheme to detect the compromised nodes by monitoring its reported data. It is a light-weighted algorithm that would incur little overhead. Considering the scalability and flexibility, hierarchical network architecture is adopted. Through intensive simulation, we verified that our WTE scheme detects misbehaved nodes accurately with very short delay.

The rest of the paper is structured as follows. In section 2, we briefly review the related malicious WSN node detection approaches. Section 3 describes our hierarchical network structure and the principle of our WTE based malicious node detection algorithm. The experiment setup and simulation results are presented in Section 4. Section 5 wraps up this paper with a discussion about efficiency and implementation issues of our solution.

2. RELATED WORK

Wireless sensor networks are often deployed in a hostile environment and work without human supervision, individual node could be easily compromised by the adversary due to the constraints such as battery lifetime, smaller memory space and limited computing capability. Security in WSN has been one of the most important topics in the WSN research community [1], [8], [22]. Here we only briefly review the reported works closely related to malicious node detection due to the limited space.

It is critical to detect and isolate the compromised nodes in order to avoid being misled by the falsified information injected by the adversary. Luo *et al*. [11] have pointed out that infrastructureless ad hoc networks rarely have a real defense mechanism against most of the attacks, including both outsider and insider attacks such as compromised node attacks. They suggested a system design like this – if one node is named trusted by certain number of its neighboring nodes, that particular node is trusted both locally and globally. However, since the system uses a minimum number of trusted nodes it is not so applicable to sensor networks where the nodes are randomly spread out. In other words, it is possible that under certain conditions nodes cannot find the minimum number of neighboring nodes in order to be named trusted.

One solution for locationized anomaly detection in a group of nodes is suggested in [4]. Every node gets the localization information from the neighboring nodes and also computes the localization information itself and compares these two values. If the difference is small enough, that node decides there is no adversary around causing the localization problem in its location.

Researchers also suggested detecting malicious node using signal strength [7]. The idea here is to depend on neighborhood monitoring of the nodes. Every sensor node monitors its surrounding and whenever a transmission signal is detected by a sensor node, it would check if the signal strength of the transmitting node is compatible with the originator node's geographical position. Even though this approach is applicable, it is not efficient in many ways. The large overhead needed for transmitting data is a problem both for sending and processing. Also it is not energy-efficient since all nodes are monitoring and processing data all the time.

The work reported in [3] is the most close to our approach. They proposed to detect malicious node by comparing its output with an aggregation value. Inspired by the Byzantine problem, our approach is more straightforward and incurs much less overhead since there is no expensive calculation involved.

Karlof and Wagner [9] suggested to construct efficient random sampling mechanisms and interactive proofs, then a user can verify that the answer given by the aggregator is a good approximation of the true value even when a fraction of the sensor nodes are compromised. Furthermore, in other fields Byzantine program is considered as an important issue. For example, in cognitive radio network, Byzantine problem in spectrum sensing is also investigated [2].

3. WEIGHTED TRUST EVALUATION TECHNOLOGY

3.1 Network Architecture

Figure 1 demonstrates the network architecture in which our weighted-trust evaluation scheme is implemented. It is a three-layer hierarchical network architecture, which consists of three types of sensor nodes similar to the architecture utilized in [20]:

- Low-power "Sensor Nodes (SN)" with limited functionality;
- Higher-power "Forwarding Nodes (FN)" that forward the data obtained form sensor nodes to upper layer;
- "Access Points (AP)", or called "Base Stations (BS)" that route data between wireless networks and the wired infrastructure.

In contrast to sensor nodes in flat ad hoc sensor networks, sensor nodes in the lowest layer of this hierarchical network do not offer multi-hop routing capability to its neighbors. A number of *Sensor Nodes* (SNs) are organized as a group and controlled by a higher layer node, the *Forwarding Node* (FN). Therefore, each sensor node only communicates with its FN and provides information such as sensor reading to its FN. FNs are located on the second layer atop the sensor node layer and offers multi-hop routing capability to SNs or other FNs. We assume the FNs are trustful and won't be compromised. We also assume the APs are trustful, otherwise the adversary can inject any data without been detected.

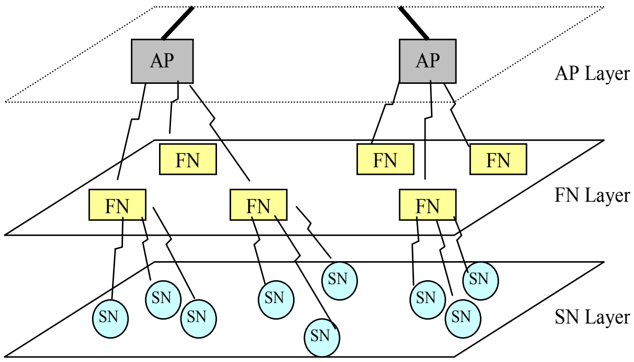


Figure 1. Architecture of the hierarchical WSN.

Each FN has two wireless interfaces, one communicates with lower layer nodes (SNs), which belong to its management, and the other connects to higher layer nodes – *Access Points* (APs).

APs are located on the highest layer in a wireless network, and have both wireless and wired interfaces. APs provide multi-hop routing for packets from SNs and FNs within radio range, in addition to routing data to wired networks. APs also have the functionality of forwarding control information from wired networks to FNs and SNs.

This hierarchical network can also be considered as a distributed information aggregation system. SNs gather information and report to its FN. Based on the information collected from SNs, FNs compute the aggregation result and commit the information to APs. However, since SNs may be compromised and report fake information, it is important for FNs to verify the correctness of the information collected from SNs. Similarly, it is also desired that APs possess the ability of verifying the committed information. Table 1 summarizes the symbolic notation used throughout this paper.

Table 1. Symbolic notations

Symbol	**Meaning**
SN	Sensor Node
FN	Forwarding node
AP	Access point
BS	Base station
W_n	Weight range
E	Aggregation result
U_n	A sensor node's output
θ	Weight penalty ratio
r_n	The ratio of sensor nodes in a cluster sending different report to the FN

3.2 Malicious Nodes Detection

As mentioned earlier, sensor nodes in sensor networks are usually deployed in hostile environments such as battlefields. Consequently a sensor node may be compromised or out of function and then provides wrong information that may mislead the whole network. This problem is called as the Byzantine problem. For example, a compromised sensor node (malicious node) can constantly report incorrect information to higher layers. The aggregator (FN or AP) in higher layer may make a wrong aggregation result due to the effect of the malicious node. It is therefore an important issue in sensor networks to detect malicious nodes in spite of such Byzantine problem.

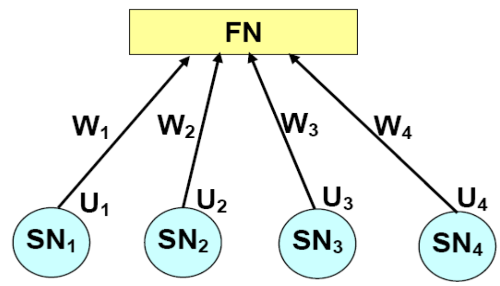


Figure 2. A weight based network for hierarchical sensor network.

As the first step toward the solution to the problem, we model it into a weight-based network as shown in Figure 2. The network is adapted in the architecture between a group of sensor nodes and their forwarding node. As shown in the figure, a weight W is assigned to each sensor node. The FN collects all information provided by SNs and calculates an aggregation result using the weight assigned to each SN:

$$E = \sum_{n=1}^{N} W_n \times U_n \tag{1}$$

Where E is the aggregation result and W_n is the weight ranging from 0 to 1. An essential concern is about the definition of sensor node's output U_n. In practice, the output information U_n may be "false" or "true" information or continues numbers such as temperature reading. Thus the definition of output U_n is usually depending on the application where the sensor network is used.

The following issue is to update the weight of each sensor node based on the correctness of information reported. Updating the weight of each sensor node has two purposes. First, if a sensor node is compromised (becomes a malicious node) and frequently sends its report inconsistent with the final decision, its weight is likely to be decreased. Then if a sensor node's weight is lower than a specific threshold, we can identify it as a malicious node. Second, the weight also decides how much a report may contribute to the final decision. This is reasonable since if the report from a sensor node tends to be incorrect, it should be counted less in the final decision.

This logic is reflected in the following equation.

$$W_n = \begin{cases} W_n - \theta \times r_n & if\,(U_n \neq E) \\ W_n & elsewise \end{cases} \tag{2}$$

Where θ is a weighted penalty ratio. When the output of a sensor node s is not consistent with the final result, its weight is reduced by the weight penalty θ multiplying r_n. The number r_n is defined as:

$$r_n = m/s \tag{3}$$

Where m is the number of nodes in the cluster sending different report to the FN, and s is the total number of nodes in the cluster under the same FN.

An optimal θ value is essential in our WTE mechanism since it affects the detection time and the accuracy of the algorithm. In addition, due to various definitions of output information (U_n) as mentioned above, the consistence determination, which decides whether a node's output is consistent with the final result, is also application-dependent. For example, it is easy to determine the consistence for a "false" or "true" output. However, for a continuous number of U_n like temperature reading, the probability distribution function could be used to determine the consistency of the output information from all sensor nodes.

Furthermore, a normalization operation as described in the following equation is used to guarantee the weight kept in the range from 0 to 1.

$$W_n = W_n / \max(W_{1,} \cdots W_N) \tag{4}$$

Based on updated weights, the forwarding node is able to detect a node as a malicious node if its weight is lower than a specific threshold.

This detection algorithm can be widely used in different types of sensor networks. For example, the number of sensor nodes can vary in the algorithm, which makes it suitable for very large and very small networks. However, the description of sensor node output and updating scaling factor which are dependent on the applied application need to be determined carefully in order to achieve efficient and high accuracy detection.

4. SIMULATION EXPERIMENTAL RESULTS

4.1 Simulation Setups

Intensive simulation experiments using MatLab were conducted to evaluate the effectiveness of our WTE based malicious nodes detection algorithm. In the simulation, the detection algorithm is deployed at a forwarding node to monitor all sensor nodes under the control of the forwarding node, and the detection is performed every cycle, which is a basic time unit of the simulation. For convenience, the output of sensor node are either as "1" (alarm) or "0" (no alarm). All simulation results were recorded after the system model reached steady state.

We assume that a sensor node is compromised randomly by the attacker at a specific probability every cycle, referred to as the attack probability, and then this malicious node keeps reporting the opposite information after compromised. For example, a malicious node always sends "alarm" while the aggregation result computed from other sensor nodes is "no alarm". Meanwhile, a normal sensor node may also send alarm when real alarm occurs. This case also occurs randomly at a different alarm probability.

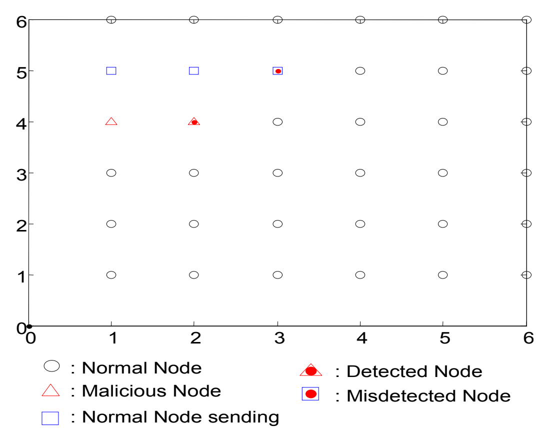


Figure 3. An example of sensor nodes deployment in the simulation.

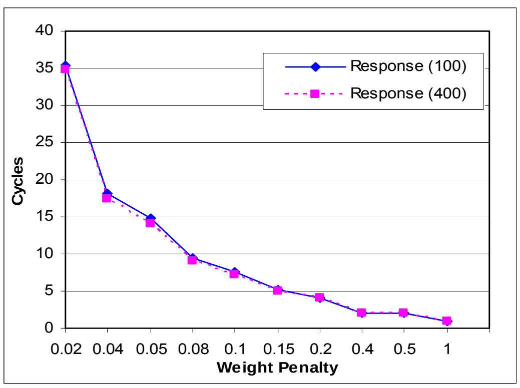
Under the assumption that sensor nodes are densely deployed to monitor certain target. In contrast to malicious nodes, if a normal node started sending alarm, its neighbor nodes would also start to send alarm after a short delay time. Furthermore, normal alarming nodes will stop sending alarms after a certain cycles. The node, which is detected or misdetected as a malicious node, is inactivated from the whole processing. The detection is terminated after 200 cycles or more than 25% of all nodes are detected as malicious nodes. Each result is calculated form an average over 1000 independent simulations.

Figure 3 shows an example of sensor nodes deployment in the simulation environment. Sensor nodes are uniformly deployed in a square plane. A sensor node may be a malicious node, a normal node, or a normal node that generating alarms.

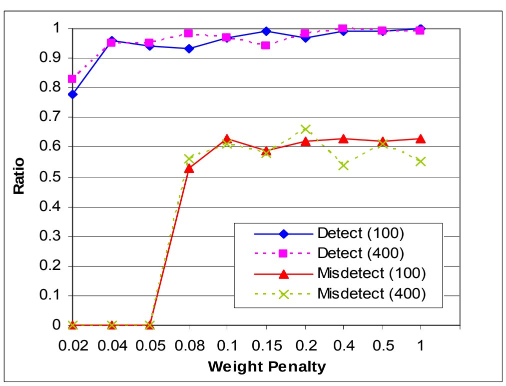
Three metrics are defined to evaluate the performance of the detection algorithm. The response time, which is the average detection cycles of correctly detected malicious nodes shows how fast malicious nodes can be detected. The Detection rate, which is the ratio of the number of detected malicious nodes and the number of total malicious nodes, indicates the effectiveness of our scheme. The third measure is misdetection ratio, which is the ratio of misdetected nodes to all detected nodes including correctly detected and misdetected nodes.

Actually these misdetected nodes consist of two parts: the number of normal nodes being treated as malicious ones and the number of malicious node being treated as normal nodes. For such a malicious node detection scheme, short response time, high detection rates are desired as well as a low misdetection ration. We studied the three metrics through simulation using different parameters.

4.2 Weight Penalty



(a) Response Time vs. Penalty Weights



(b) Detection Accuracy vs. Penalty Weights

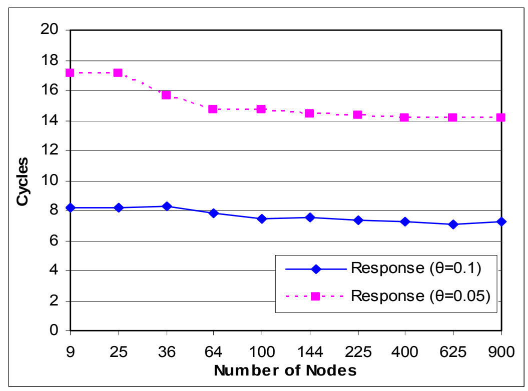
Figure 4. Impact of various Penalty Weights on system performance.

The first simulation is to find an optimal weight penalty for the detection algorithm. The attack probability and alarm probability are both 0.04. The number of cycles that normal nodes send alarms and wait to stop alarms is 10 cycles. A threshold (0.4) is also set for detection determination as mentioned earlier.

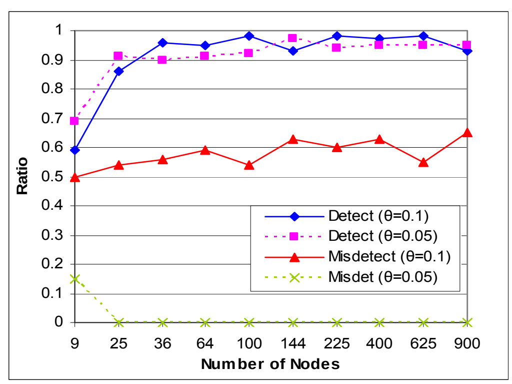
Figure 4 illustrates the results with weight penalties varying from 0.02 to 1.0 the number of sensor nodes are change from 100 nodes to 400 nodes. The increasing weight penalty reaches a shorter response time, and improves the detection ratio. Intuitively the penalty value reveals the sensitivity of our detection results against the variation in reported data. However, the misdetection ratio also increases as weight penalty increasing, especially after the penalty ratio becomes 0.08 and greater. Considering the tradeoffs among response time, detection rate and misdetection rate comprehensively, it is reasonable to set the weight penalties values in the range of (0.04-0.1).

4.3 Scalability

Using weight penalties 0.1 and 0.05, we further evaluated the algorithm with various numbers of nodes as shown in Figure 5. The parameters for this experiment are the same as the first experiment.



(a) Response Time vs. Number of Nodes



(b) Detection Accuracy vs. Number of Nodes

Figure 5. Illustration of the system scalability.

The response time, detection, and misdetection ratios are pretty stable while we increased the number of nodes from 9 to 900, particularly when the number of nodes is greater than 64. This result implies that our WTE based detection algorithm has very nice scalability as it works well under variant network sizes without losing much performance. Especially if the size of network becomes large enough, for example, greater than 64, the network size almost has no influence on the performance.

Figure 5 also demonstrates the impact of the selection of penalty weight θ. When a larger value is chosen ($\theta = 0.1$), the system can detect malicious node faster and more accurately comparing to using smaller value ($\theta = 0.05$) as shown in Fig. 5(a) and the upper two curves in Fig. 5(b).

However, such a faster response is achieved with the cost of higher misdetection rate as shown by the lower two curves in Fig. 5(b). This verifies the tradeoff among detection performance and misdetection ratio, and shows that the system operator can adjust the sensitivity of the penalty weight parameter θ according to the requirements in different applications.

4.4 Attack Probability

Finally, the performance at various attack probabilities is evaluated with weight penalty 0.05 for 100 nodes and 400 nodes cases. The attacking probability is defined as the ratio of malicious nodes among total number of sensor nodes in the network that is assumed could be compromised. It describes the intensity of false data that the adversary injects into the network.

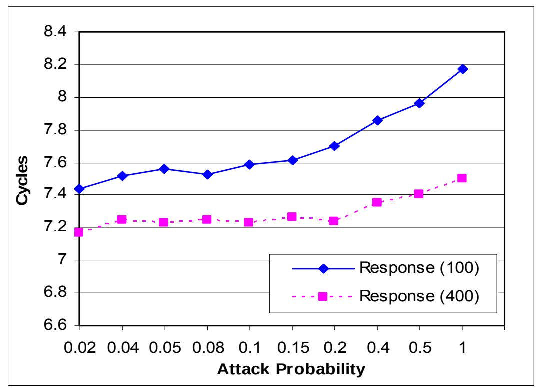
As indicated in the research of Byzantine General Problem [10], when the number of malicious nodes is larger than the number of legitimate nodes, the loyalty generals cannot figure out who is the rebelled one; Furthermore, when there is not any authentication mechanism applied, the number of rebel generals has to be less than 1/3 of the total number of generals if the loyal generals want to reach an agreement on correct activity.

In our problem, similarly, if the number of compromised nodes is larger than 25% of the total nodes, we may not be able to detect the “bad guys” accurately. The upper bond of the amount of compromised nodes in our simulation is 30% of the total number of nodes. Therefore, the attack probability of 1 implies that there are 25% of the sensor nodes have been compromised.

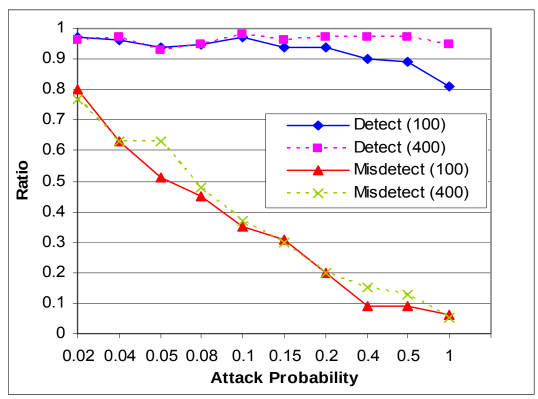
We evaluated the performance using the response time, detection, and misdetection ratios as shown in Figure 6. The increasing attack probability means that there are more nodes being compromised and falsified data are inserted.

As illustrated in Fig. 6(a), the response time slightly increases with attack probability increasing. This makes sense that as more malicious nodes appear, the aggregated data is affected more by the falsified data. While there are only small changes observed in detection ratios, the misdetection ratio decreases largely as the growth of the attack probabilities, as shown in Fig. 6(b). This is partially due to the increasing number of malicious nodes that makes the false positive rate smaller.

Based on the results reported above, the response time, detection, and misdetection ratios are stable in the cases large number of nodes and high compromise probability. It demonstrates that the proposed detection algorithm is efficient for both large networks and high attack probability conditions. The experiment results also show that the performance of the detection algorithm is largely dependent on parameters studied above.



(a) Response Time vs. Compromise Probability



(b) Detection Accuracy vs. Compromise Probability

Figure 6. System performance under different compromise probabilities.

5. CONCLUSIONS

In this paper, we proposed a novel weighted-trust evaluation based scheme to detect compromised or misbehaved nodes in wireless sensor networks. The basic idea is that FNs give trust values to each of the nodes in the cluster, if a node sends meaningless/wrong information which implies that a node has been compromised or out is of function, the FN directly lowers that node's trust level. It is much easier and less complex to keep track of the nodes and it is harder to compromise most of the node unless an attacker compromises the base stations.

With a very good scalability, our approach is applicable to both small size WSNs and WSNs with larger number of nodes. The only difference to apply it to larger size WSNs is to increase the number of FNs. Essentially, it could be treated as a node-clustering problem.

Although there are couples of research works reported addressing the malicious node detection problem in WSNs, it is difficult to compare the performance between each other. As introduced in section 2, the design assumptions and the experiments environments are very different. Particularly, lack of a comparable benchmark makes it meaningless to compare the results, i.e. detection rate.

Our approach is based on the assumption that base stations are trusted. In fact, if the adversary can gain control over the base stations, he/she can do any possible attack against the WSN. This is an interesting open problem, however it is beyond the scope of this paper. Another critical assumption is that the majority of the sensor nodes are working properly. If the number of compromised nodes is more than the number of normal nodes, the legal nodes will be reported as malicious one and being isolated.

Actually, in this paper we have reported merely some preliminary results, which verified the correctness and effectiveness of our solution. More detailed analysis regarding the performance of our scheme will be studied in the ongoing research and more questions to be answered. For instance, how is the impact of distribution of the 25% malicious nodes against the performance of weighted-trust evaluation? What is the behavior of our detection scheme if the ratio of malicious nodes beyond 1/3 of the sensor nodes?

In our progressive efforts, we are studying the deployment of FNs and the influence of different densities of FNs on the performance. In addition, we are setting up a testbed consisting of more than 64 sensor nodes. That may allow us to investigate the differences between the simulation experiments and what happens in real world when real physical nodes are in use.

ACKNOWLEDGEMENT

We'd like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] E. Ayday, F. Delgosha, and F. Fekri, "Location-Aware Security Services for Wireless Sensor Networks using Network Coding," *Infocom*, May 2007.

[2] R. Chen, J. M. Park, and K. Bian, "Robust Distributed Spectrum Sensing in Cognitive Radio Networks," *Technical Report TR-ECE-06-07*, Dept. of Electrical and Computer Engineering, Virginia Tech., July 2006.

[3] D.-I. Curiac, O. Banias, F. Dragan, C. Volosencu, and O. Dranga, "Malicious Node Detection in Wireless Sensor Networks Using an Autoregression Technique," *the 3rd International Conference on Networking and Services* (ICNS'07), June 19 – 25, 2007, Athens, Greece.

[4] W. Du, L. Fang, and P. Ning, "LAD: Localization Anomaly Detection for Wireless Sensor Networks," *the 19th International Parallel and Distributed Priocessing Symposium* (IPDPS'05), April 3 – 8, 2005, Denver, Colorado, USA.

[5] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *MOBICOM*, August 1999

[6] Y. Hu, A. Perrig, and D. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks," *IEEE INFOCOM*, 2003

[7] W. Junior, T. Figueriredo, H.-C. Wong, and A. Loureiro, "Malicious Node Detection in Wireless Sensor Networks," *the 18th International Parallel and Distributed Priocessing Symposium* (IPDPS'04), April 26 – 30, 2004, Santa Fe, Nex Mexico, USA.

[8] C. Karlof, N. Sastry, and D. Wagner, "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks," *ACM Sensys*, November 2004.

[9] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," *Journal of Ad Hoc Networks*, Elsevier, 2003

[10] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Transactions on Programming Languages and Systems*, Vol. 4, No. 3, July 1982.

[11] Haiyun Luo, Petros Zerfos, Jiejun Kong, Songwu Lu, Lixia Zhang, "Self-securing Ad Hoc Wireless Networks," *IEEE ISCC (IEEE Symposium on Computers and Communications)* 2002, Italy.

[12] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil Attack in Sensor Networks: Analysis and Defense," *International Symposium on Information Processing in Sensor Networks*, Vol. 1(2004).

[13] B. Przydatek, D. Song, and A. Perrig, "SIA: Secure Information Aggregation in Sensor Networks," *Proceedings of the 1st international conference on*

Embedded networked sensor systems, November 05-07, 2003, Los Angeles, California, USA.

[14] S. D. Servetto, "From 'small Sensor Networks' to 'Sensor Networks'", *EmNets 2006*, May 2006.

[15] B. Sun, K. Wu, and U. Pooch, "Secure Routing against Black-hole Attack in Mobile Ad Hoc Networks," in *Proceedings of Communications and Computer Networks*, 2002.

[16] M. Tubaishat and S. Madria., "Sensor Networks: an Overview," *IEEE Potentials*, 22, 2, 20-23, April 2003.

[17] M.A.M. Vieira, D.C. da Silva Jr., C.N. Coelho Jr., and J.M. da Mata., "Survey on Wireless Sensor Network Devices," *Emerging Technologies and Factory Automation* (ETFA03), September 2003.

[18] W. Ye, F. Silva, and J. Heidemann, "Ultra-Low Duty Cycle MAC with Scheduled Channel Polling", in *Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems* (SenSys), Boulder, Colorado, USA, November, 2006.

[19] Y. Yu, B. Krishnamachari, and V.K. Prasanna, "Energy-Latency Tradeoffs for Data Gathering in Wireless Sensor Networks," *IEEE Infocom'04*

[20] S. Zhao, K. Tepe, I. Seskar and D. Raychaudhuri, "Routing Protocols for Self-Organizing Hierarchical Ad-Hoc Wireless Networks," *Proceedings of the IEEE Sarnoff Symposium*, Trenton, NJ, March 2003.

[21] L. Zhou and Z. Haas, "Securing Ad Hoc Networks," *IEEE Network Special Issue on Network Security*, 13, 6, 24-30, November 1999.

[22] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," *CCS'03*, October 2003.

BIOGRAPHY

Idris M. Atakli received his M.S. degree in Electrical Engineering from the State University of New York (SUNY), Binghamton in 2007. He is currently pursuing his Ph.D. degree in the Department of Electrical and Computer Engineering at SUNY - Binghamton. His research interests include information security, digital watermark applications, and computer network systems. Mr. Atakli could be reached at iatakli1@binghamton.edu.

Hongbing Hu received his B.E. in computer science from Jilin University, Changchun, China, and B.E. in information and network science from Chiba Institute of Technology, Narashino, Japan both in 2001. He received his M.S. degree in information science from Tohoku University, Sendai, Japan in 2003. He is currently pursuing his Ph.D. degree in the Department of Electrical and Computer Engineering at the SUNY - Binghamton. His research interests include speech analysis, pattern recognition, speech coding, and network systems. Mr. Hu could be reached at hhu1@binghamton.edu.

Yu Chen received the MS and PhD degree in Electrical Engineering from the University of Southern California (USC) in 2002 and 2006, respectively. He is an assistant professor of electrical and computer engineering at SUNY - Binghamton. His research interest includes network security, Security and privacy in distributed systems and pervasive computing environments, Internet infrastructure security, and reconfigurable hardware based security solutions. He is a member of the ACM, the IEEE and the SPIE. Dr. Chen could be reached at ychen@binghamton.edu.

Wei-Shinn Ku received the Ph.D. degree in computer science from the University of Southern California (USC) in 2007. He also obtained both the M.S. degree in computer science and the M.S. degree in Electrical Engineering from USC in 2003 and 2006 respectively. He is an assistant professor with the Department of computer science and software engineering at Auburn University. His research interests include spatial and temporal data management, mobile data management, geographic information systems, and security & privacy. He has published more than 30 research papers in refereed international journals and conference proceedings. He is a member of the ACM and the IEEE. Dr. Ku could be reached at weishinn@auburn.edu.

Zhou Su received Ph.D from Waseda University, Japan in 2003. He also received the B.S and M.S from Xi'an Jiaotong University, China, in 1997 and 2000 respectively. He is an Assistant Professor with Department of Computer Science at Waseda University. His research interests include Network Traffic analysis, Internet Architecture, Contents Delivery, Mobile Multimedia, P2P, Overlay Networks, and new applications on WWW. Dr. Su could be reached at zhousu@asagi.waseda.jp.