

---

## Weighted trust evaluation-based malicious node detection for wireless sensor networks

---

Hongbing Hu and Yu Chen\*

State University of New York – Binghamton  
Binghamton, NY 13902, USA

E-mail: hhu1@binghamton.edu

E-mail: ychen@binghamton.edu

\*Corresponding author

Wei-Shinn Ku

Auburn University  
Auburn, AL 36849, USA

E-mail: weishinn@auburn.edu

Zhou Su

Waseda University  
Ohkubo 3–4–1, Shinjyuku, Tokyo 169–8555, Japan

E-mail: zhousu@asagi.waseda.jp

Chung-Han J. Chen

Tuskegee University  
Tuskegee, AL 36088, USA

E-mail: jchen@tuskegee.edu

**Abstract:** Deployed in a hostile environment, the individual Sensor Node (SN) of a Wireless Sensor Network (WSN) could be easily compromised by an adversary due to constraints such as limited memory space and computing capability. Therefore, it is critical to detect and isolate compromised nodes in order to avoid being misled by the falsified information injected by adversaries through compromised nodes. However, it is challenging to secure the flat topology networks effectively because of the poor scalability and high communication overhead. On top of a hierarchical WSN architecture, a novel algorithm based on *Weighted Trust Evaluation* (WTE) to detect malicious nodes for hierarchical sensor networks is proposed in this paper. The hierarchical network can reduce the communication overhead among SNs by utilising clustered topology. The proposed algorithm models a cluster of SNs and detects malicious nodes by examining their weights that represent the reliability of SNs. Through intensive simulations, the accuracy and effectiveness of the proposed detection algorithm are verified.

**Keywords:** wireless sensor networks; WSNs; network security; hierarchical topology; malicious node detection.

**Reference** to this paper should be made as follows: Hu, H., Chen, Y., Ku, W-S., Su, Z. and Chen, C-H.J. (2009) 'Weighted trust evaluation-based malicious node detection for wireless sensor networks', *Int. J. Information and Computer Security*, Vol. 3, No. 2, pp.132–149.

**Biographical notes:** Hongbing Hu received his BE in Information and Network Science from Chiba Institute of Technology, Narashino, Japan in 2001. He received his MS Degree in Information Science from Tohoku University, Sendai, Japan in 2003. He is currently pursuing his PhD Degree in the Department of Electrical and Computer Engineering at the State University of New York (SUNY) at Binghamton, USA. His research interests include speech analysis, pattern recognition, speech coding and network security. He is a member of the IEEE.

Yu Chen received his MS and PhD Degrees in Electrical Engineering from the University of Southern California (USC), USA in 2002 and 2006, respectively. He is an Assistant Professor of Electrical and Computer Engineering at the State University of New York (SUNY) at Binghamton. His research interests include 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.

Wei-Shinn Ku received his PhD Degree in Computer Science from the University of Southern California (USC), USA in 2007. He also obtained both his MS Degree in Computer Science and his MS 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 and 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.

Zhou Su received his PhD Degree from Waseda University, Japan in 2003. He also received his BS and MS from Xi'an Jiaotong University, China, in 1997 and 2000 respectively. He is an Assistant Professor with the 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 the WWW.

Chung-Han J. Chen received his PhD Degree in Computer Engineering from the University of Louisiana at Lafayette, USA. He is currently an Associate Professor of Computer Science at Tuskegee University. His research interests are in information and network security, operating systems, computer architecture and VLSI design. He has been a member of the IEEE Computer Society since 1985. He served as the workshop chair of the 2008 ACM SE conference.

---

## 1 Introduction

Recent achievements in the areas of *Micro-Electro-Mechanical Systems* (MEMS) and low power integrated electronic devices have led to the development and wide application of *Wireless Sensor Networks* (WSNs) (Estrin *et al.*, 1999; Servetto, 2006; Tubaishat and Madria, 2003). WSNs consist of very small devices, called Sensor Nodes (SNs). These SNs are battery powered and equipped with integrated sensors, a data-processing unit, a small storage memory, and short-range radio communication (Vieira *et al.*, 2003). Typically, these sensors are randomly deployed in a field, forming an unattended wireless network. The objective of these SNs is to collect data from the field, partially aggregate data, and send aggregated data 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 monitoring.

In contrast to traditional wireless networks, special security and performance issues have to be carefully considered for sensor networks (Zhou and Haas, 1999). For instance, 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 damage should be minimised. In other words, compromising a single SN or a few SNs should not be able to crash the entire network.

Another concern is about energy efficiency of SNs. In a sensor network, each SN 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 (Yu *et al.*, 2004). Especially, the number of message transmissions and the amount of expensive computation should be as small as possible.

In fact, there are a number of attacks that an attacker can launch against a WSN. For instance, HELLO flooding attacks (Karlof and Wagner, 2003), sink hole attacks (Karlof and Wagner, 2003), Sybil attacks (Newsome *et al.*, 2004), black hole attacks (Sun *et al.*, 2002), worm hole attacks (Hu *et al.*, 2003), or DDoS attacks (Du *et al.*, 2005) are well-known options for attackers. These attacks lead to anomalies in network behaviours that are detectable in general. There are some solutions reported to detect these attacks by monitoring the anomalies (Karlof and Wagner, 2003).

In this paper, we address an even trickier attacking scenario. After gaining the control over a number of SNs in a network, an attacker performs an insider attack, sending falsified information through the compromised nodes instead of simply destroying these nodes. The purpose of this insider attack is to mislead the operator of the network by using 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 unnecessary casualties.

We proposed a *Weighted Trust Evaluation* (WTE) based algorithm to detect compromised nodes by monitoring the data that the nodes reported. This is a light-weighted algorithm with little overhead. The algorithm is based on a hierarchical topology network for reducing the communication overhead among SNs. Base stations including Forwarding Nodes (FNs) and Access Points (APs), playing different roles in

this hierarchical network, are also introduced. The proposed algorithm models a cluster of SNs under the control of a FN and detects malicious nodes by examining their weights. These weights are assigned to SNs, representing the reliabilities of SNs.

The rest of the paper is structured as follows. In Section 2, we briefly review several related malicious node detection approaches. Section 3 describes a hierarchical network structure and the principle of the 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 the effectiveness and implementation issues of the algorithm.

## 2 Related work

WSNs are often deployed in a hostile environment and work without human supervision. Individual node could be easily compromised by an adversary due to the constraints such as smaller memory space and limited computing capability. Security has been one of the most important topics in the research community of sensor networks (Ayday *et al.*, 2007; Karlof *et al.*, 2004; Zhu *et al.*, 2003). In this section we briefly review some reported works closely related to malicious node detection.

It is critical to detect and isolate the compromised nodes in order to avoid being misled by the falsified information injected by an adversary. Researchers (Luo *et al.*, 2002) 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 that if one node is named trusted by certain number of its neighbouring 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 are not able to find the minimum number of neighbouring nodes to be named trusted.

One solution for localisation anomaly detection in a group of nodes is suggested (Du *et al.*, 2005). Every node obtains the localisation information from its neighbouring nodes, and then computes the localisation information itself and compares these two values. If the difference is small enough, this node concludes there is no adversary around causing the localisation problem in its location.

Researchers also suggested detecting malicious nodes using signal strength (Junior *et al.*, 2004). The basic idea is to depend on neighbourhood monitoring of SNs. Every SN monitors its surrounding and whenever a transmission signal is detected, it will examine whether 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 major problem for both sending and processing. Also it is not energy-efficient since all nodes are monitoring and processing data all the time.

The work reported in Curia *et al.* (2007) is the closest to our approach. They proposed to detect a 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 no expensive calculation is involved.

Karlof and Wagner (2003) suggested to construct efficient random sampling mechanisms and interactive proofs, thus a user can verify that the answer given by the aggregator is a good approximation of the true value even when a fraction of SNs are compromised. Apart from WSNs, the Byzantine problem is also considered as an important issue in other fields. For instance, in the research of cognitive radio network (Chen *et al.*, 2006), the Byzantine problem was investigated to enhance the robustness of distributed spectrum sensing against terminals experiencing Byzantine failure.

From the perspective of system level security on top of cooperative behaviour among individual SNs, reputation-based frameworks have been developed (Ganerwal and Srivastava, 2004; Sun *et al.*, 2006). Each SN maintains reputation values of its peers, based on that a node can make decision whether its peers are trustworthy. Hence, malicious or misbehaviour nodes will be isolated from the cooperation group. This approach works well in distributed systems where cooperation among agents is desired to improve the system performance. However, the storage and computing power constraints tremendously limit the SNs' capability of handling the reputation maintenance and calculation jobs.

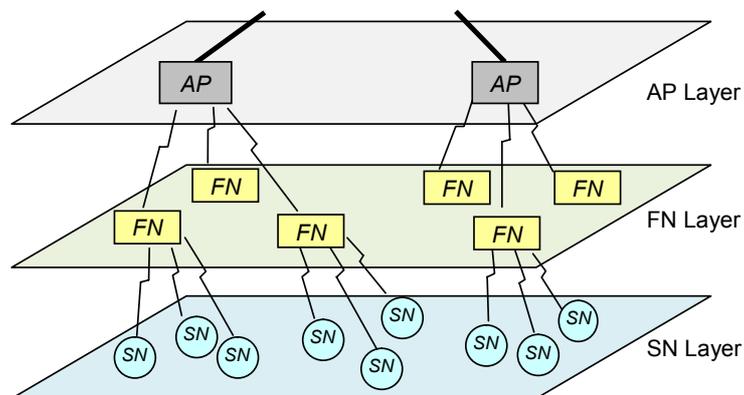
### 3 Weighted Trust Evaluation (WTE) based algorithm

#### 3.1 Hierarchical network architecture

Figure 1 depicts the network architecture in which the WTE based detection algorithm is implemented. This architecture is similar to the one utilised in Zhao *et al.* (2003) that is based on a three-layer hierarchical topology, consisting of the following three types of nodes:

- 1 *Sensor Nodes (SNs)*: common nodes in a WSN with limited functionality
- 2 *Forwarding Nodes (FNs)*: more powerful nodes that forward the data obtained from SNs to the upper layer
- 3 *Access Points (APs)*: nodes that route data between wireless networks and the wired infrastructure.

**Figure 1** Architecture of the hierarchical WSN (see online version for colours)



A number of SNs are organised as a group and controlled by a higher layer node: FN. Therefore, in contrast to SNs in flat *ad hoc* sensor networks, SNs in this hierarchical network are not equipped with multi-hop routing function to their neighbour nodes. Each SN communicates only with its FN including sending information to and receiving information from FNs.

FNs in the middle layer on top of the SN layer offer multi-hop routing function. Each FN has two wireless interfaces: one communicates with lower layer SNs belonging to its management, and the other connects to a higher layer node: AP.

APs located on the highest layer have wireless and wired interfaces, providing multi-hop routing for all sensor and FNs within the radio range as well as routing data to the wired network. Moreover, APs have the functionality of forwarding control information from the wired network to forwarding and SNs. In this paper, we assume FNs are trustful and not compromised by any attack. We also assume APs are trustful, otherwise the adversary can inject any data without been detected.

Meanwhile, this hierarchical network can be considered as a distributed information aggregation network (Przydatek *et al.*, 2003). Based on the information reported by SNs, FNs compute an aggregation information and commit the information to APs. Since SNs may be compromised and report falsified information, it is important for FNs to verify the correctness of the information. Similarly, it is also desired that APs possess the ability of verifying the committed information.

Table 1 summarises the symbolic notations used throughout this paper to aid in explanation of the WTE based algorithm.

**Table 1** Symbolic notations

<i>Symbol</i>	<i>Meaning</i>
SN	Sensor node
FN	Forwarding node
AP	Access point
$W_n$	Weight value of a sensor node
$E$	Aggregation result
$U_n$	Output of a sensor node
$\theta$	Weight penalty
$\beta$	Weight value recovery rate
$r_n$	A factor reflecting the coincidence of a sensor node with its neighbouring nodes
$N$	Number of sensor nodes under the same forwarding node
$N_s$	Number of the neighbouring nodes of a sensor node
$N_m$	Number of nodes sending different report among the neighbouring nodes
$t_h$	Recovery time, time length for weight value recovery

### 3.2 Malicious node detection

As mentioned previously, SNs in WSNs are usually deployed in hostile environments such as battlefields. Consequently a SN may be compromised or out of function and then provides incorrect information to mislead the whole network. This problem is called

as the Byzantine problem (Lamport *et al.*, 1982). For instance, a compromised SN (malicious node) may frequently report incorrect information to higher layers. The aggregator (FN) is thus not able to obtain correct aggregation information due to the effect of this malicious node. Thus, the detection of malicious nodes becomes an important issue in WSNs.

At the first step, a weight based network that applies WTE is adapted for a group of SNs and its FN. As shown in Figure 2, a weight  $W_n$  is assigned to each SN to represent the reliability of the node. In the process of aggregating the information sent by the SNs, the FN utilises the weights and calculates the aggregation result as follows:

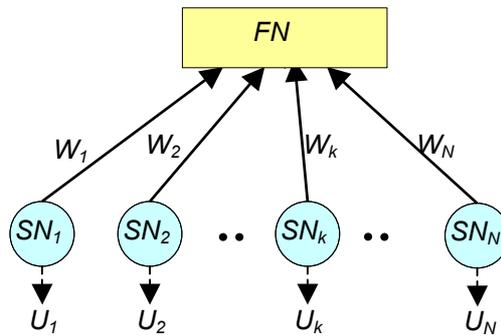
$$E = \sum_{n=1}^N W_n \times U_n, \tag{1}$$

where:

- $E$  = the aggregation result
- $W_n$  = the weight with the value ranging from 0 to 1
- $N$  = the number of SNs in the group.

One concern is about the output  $U_n$  definition of SNs. In practice, the output information  $U_n$  may be binary information (*e.g.*, ‘false’ or ‘true’) or continuous numerical values (*e.g.*, temperature reading). Thus the definition of the output  $U_n$  is application dependent.

**Figure 2** Weight trust evaluation for malicious node detection (see online version for colours)



Then, the next issue is to update the weight value based on the correctness of information reported by each SN. Updating the weight of each SN has two purposes. First, if a node is compromised (becomes a malicious node) and frequently sends its report inconsistent with the aggregation result, its weight value is going to be decreased. Once the weight value of a SN is lower than a threshold, the node can be considered as a malicious node. Second, the weight is used to represent how much the information of a node may contribute to the aggregation result. This is reasonable since if the report from a node tends to be incorrect, it should be counted less in the calculation of the aggregation result.

This logic is reflected in the following equation:

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

where  $\theta$  is a weighted penalty. When the output of a SN is not consistent with the aggregation result  $E$ , its weight value is decreased by the weight penalty  $\theta$  multiplying a factor  $r_n$ . The  $r_n$  reflecting the coordination of the node with its neighbouring nodes is defined as:

$$r_n = N_m / N_s \quad (3)$$

where  $N_s$  is the total number of the neighbouring nodes, and  $N_m$  is the number of nodes sending different report among the neighbouring nodes.

Furthermore, a normalisation operation as described in the following equation is used to keep weight values of SNs in the range of [0, 1].

$$W_n = W_n / \max(W_1, \dots, W_N). \quad (4)$$

Finally, the weight of each SN is periodically examined by the FN. If the weight value is lower than a threshold, the node is identified as a malicious node and isolated from the network.

This WTE based detection algorithm can be widely used in different types of sensor networks. The number of SNs could be varied, making the algorithm suitable for arbitrary size networks. Note that the parameters including the weight penalty and threshold are dependent on applications and need to be determined carefully for achieving an effective and accurate detection. For instance, the weight penalty could greatly change the detection time and the accuracy of the algorithm.

### 3.3 Weight value recovery

As presented in Section 3.2, the weight value of a SN is decreased once it is detected reporting incorrect information. However, this incorrect information may be merely due to a temporary interruption in communication channel, the SN is neither compromised nor out-of-function. It is not desired to keep the weight values of such nodes low permanently. Thus, a mechanism is needed to recover weight values of SNs if they work normally after that disturbance. For this purpose, an adaptive weight value recovery algorithm is proposed in this section.

The rationales of this algorithm are considered as follows. If a SN has been out of function, the data from it will always mismatch the aggregation result. Also, if a node has been compromised by the adversary, at least it needs to report falsified information for certain length of time if it intends to mislead the operator of the sensor network. Therefore, whether it is time to recover the weight value depends on the behaviour of a node during the past certain period of time.

This logic is reflected in the following equation, if the weight value  $W_n < 1$ , this value is updated as:

$$W_n = \begin{cases} \min[W_n + \beta \times (1 - r_n), 1] & \text{if } (U_n \approx E \text{ and } t_c \geq t_h) \\ W_n & \text{elsewise} \end{cases}, \quad (5)$$

where:

$\beta$  = a weight recovery rate

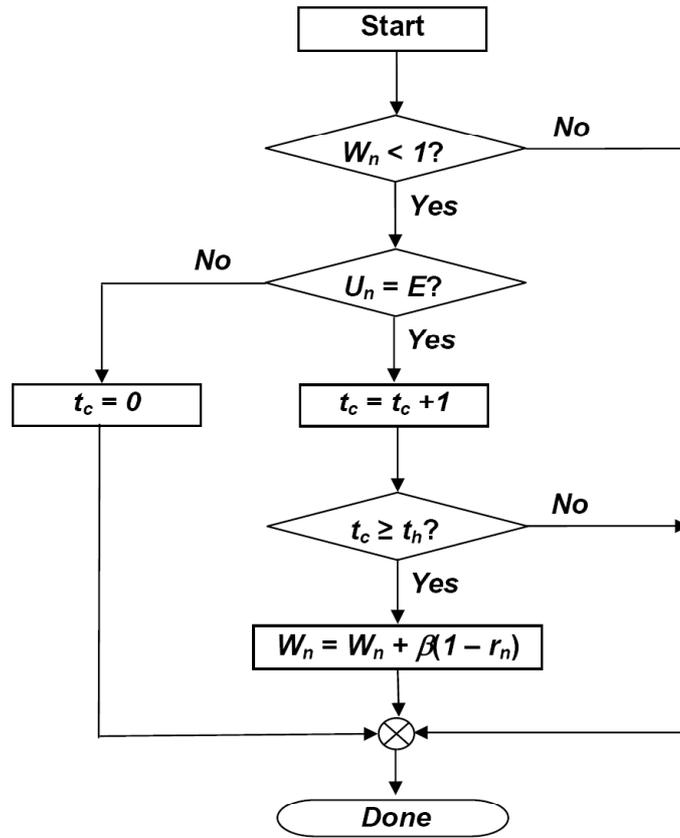
$t_c$  = the time in which the SN behaves correctly

$t_h$  = (preset threshold) the required length of time for weight value recovery, named as recovery time.

Only if the SN has been behaving correctly longer than the recovery time  $t_h$ , its weight value is increased. The parameter  $r_n$  is defined as Equation (3) in Section 3.2.

In experiments reported in this paper, we assume that each SN reports to the FN periodically, both  $t_h$  and  $t_c$  are defined in number of period cycles. Once a node's weight value become lower than 1, its past behaviour is examined and compared with the preset threshold  $t_h$ . If the node has worked normally long enough, its weight value is recovered according to Equation (5). The flowchart of the weight recovery algorithm is presented in Figure 3.

**Figure 3** Flowchart of the weight value recovery mechanism



In this algorithm, the set of key parameters including the weight recovery rate  $\beta$  and the recovery time threshold  $t_h$  determine the performance of the system. If  $\beta$  is set too large, or the threshold  $t_h$  is set too low, a malicious node can easily regain its trust value, which was reduced by the algorithm in Section 3.2. On the other hand, if the threshold  $t_h$  is too high, an innocent node may be punished for a long time unnecessarily. However, there is no theoretical model to describe the impact precisely. Therefore, we investigate their impact through intensive simulation experiments to choose the optimal values. The simulation experimental results are reported in Section 4.3 in detail.

## 4 Simulation experimental results

### 4.1 Simulation setup

Intensive simulation experiments using MATLAB have been conducted to evaluate the effectiveness of the WTE based malicious node detection algorithm. In the simulation, the detection algorithm is deployed at a FN to monitor all SNs under the control of this FN. The detection is performed every cycle, which is used as a basic time unit of the simulation. For convenience, the output of SN is either as '2' (alarm) or '1' (no alarm).

Since a continuous value is used in Equation (1), it may be a concern that whether the results under such a particular subset of parameters could hold for general cases. For the purpose of this paper, it is enough to verify the correctness and effectiveness of the WTE idea. Essentially, the issue has been simplified by using '2' or '1' is the resolution that our monitoring nodes possess. We are investigating this problem in our ongoing efforts.

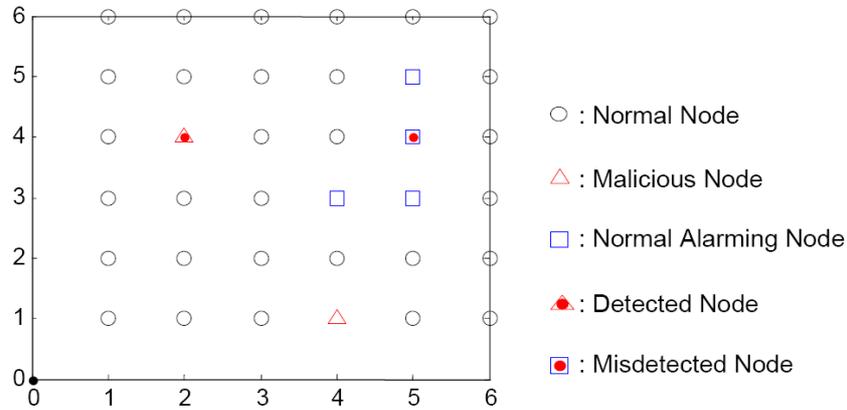
Assume that a SN is compromised randomly by an attacker in each cycle at a preset probability, referred to as the attack probability, and the malicious node keeps reporting the opposite information once compromised. For instance, a malicious node always sends 'alarm' while the aggregation result computed from all SNs is 'no alarm'. Meanwhile, a normal SN may also send alarm when real alarm occurs. This case also occurs randomly at a different probability called the alarm probability.

Under the assumption that SNs are densely deployed to monitor certain target. In contrast to malicious nodes, if a normal node started sending alarms, some of its neighbour nodes will be triggered to start sending alarms after a short delay time. Furthermore, normal alarming nodes stop sending alarms after a certain cycles. The node, which is detected or misdetected as a malicious node, is inactivated from the whole processing except when the weight value recovery mechanism is incorporated.

Figure 4 shows an example of SN deployment in the simulation environment. SNs are uniformly deployed in a square plane. A SN may be a malicious node, a normal node, or a normal node that generating alarms.

As indicated in the research of Byzantine General Problem (Lamport *et al.*, 1982), when the number of betrayed generals exceeds one third of the total number, the loyal generals cannot achieve a good decision. In our problem, similarly, if the number of malicious nodes is larger than 33% of the total nodes, it could be difficult to detect the malicious nodes accurately. Therefore, the detection is terminated when more than 25% of all nodes are detected as malicious nodes or reaching 200 simulation cycles. Each result is calculated form an average over 1000 independent simulations. As the scenarios where the number of malicious nodes exceeds 25% or even beyond 33%, we leave that for our further studies.

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 ratio*, which is the ratio of the number of detected malicious nodes to the number of total malicious nodes, indicates the effectiveness of our algorithm. The third one is *misdetection ratio*, which is the ratio of misdetected nodes to all detected nodes including correctly detected and misdetected nodes. Note that these misdetected nodes actually consist of two categories: normal nodes being treated as malicious ones and malicious node being treated as normal nodes.

**Figure 4** An example of sensor node deployment in the simulation (see online version for colours)

For such a malicious node detection algorithm, short response time, high detection ratio are desired as well as a low misdetection ratio. In the following sections, we will study the impact of several key parameters used in the algorithm, and investigate the scalability and robustness of the algorithm.

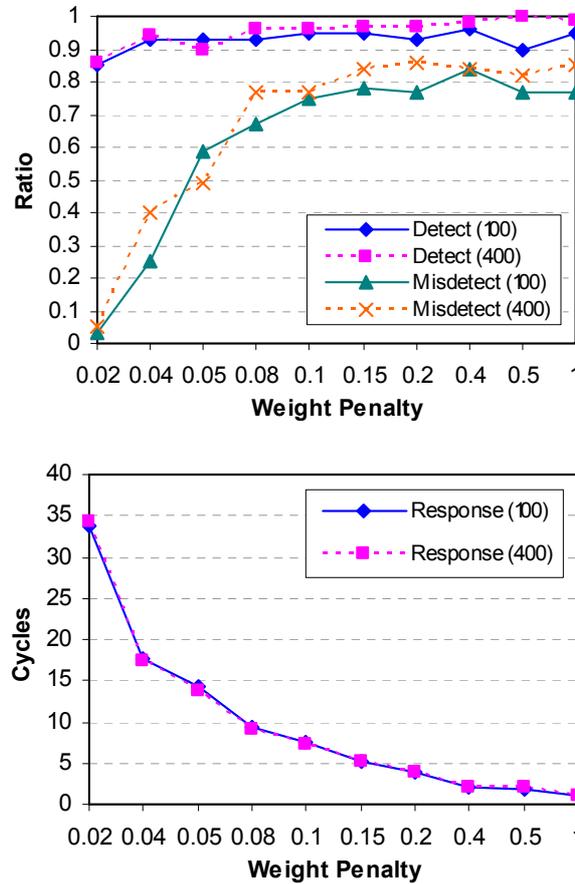
#### 4.2 Weight penalty

The first simulation was conducted to explore an optimal weight penalty value for the detection algorithm. The attack and alarm probabilities are both set to 0.04. The number of cycles between normal nodes start and stop sending alarms is 10. A threshold (0.4) is also set for malicious node detection as mentioned in Section 3.2. The weight recovery mechanism is not employed in this simulation for convenience.

Figure 5 shows the results with weight penalty value varying from 0.02 to 1.0 in 100 and 400 SNs cases. The larger weight penalty value results in a shorter response time, and a slightly better detection ratio. Intuitively the penalty value reveals the sensitivity of our detection results against the variation in reported data. For instance, when a large value is chosen ( $\theta=0.1$ ), the algorithm is able to detect malicious node twice faster and approximately 10% more accurately comparing to using  $\theta=0.05$ . However, such a fast response is achieved with the cost of high misdetection ratio. The misdetection ratio increases as weight penalty increasing, especially after the penalty becomes 0.08 and greater.

These results have verified the tradeoff between detection performance and misdetection ratio, and shown that the penalty weight value need to be adjusted according to the requirements in different applications. Considering these factors comprehensively, we used 0.1 as the weight penalty in the following experiments.

**Figure 5** Detection accuracies (top) and response time (bottom) with various weight penalty values (see online version for colours)

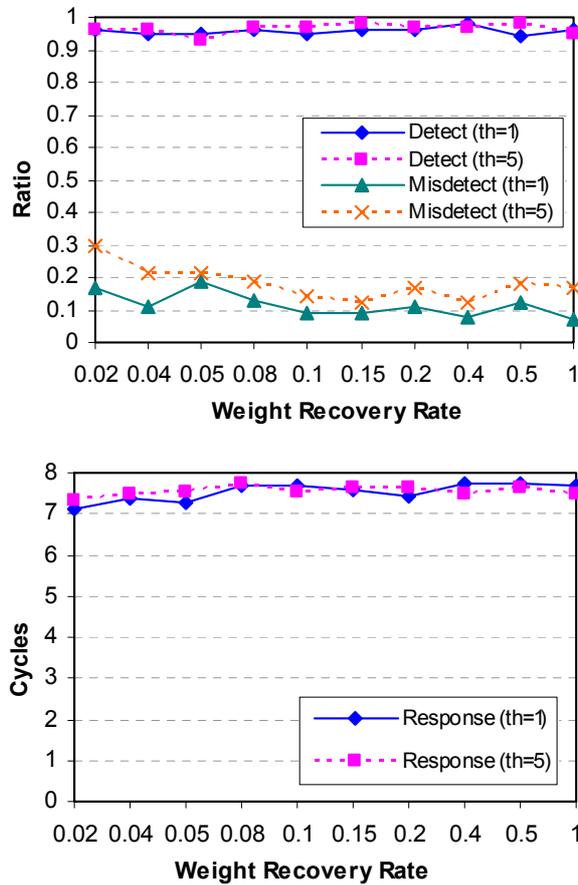


### 4.3 Weight value recovery

Then, we incorporated the weight recovery mechanism in the detection algorithm, and evaluated its performance with variant selections of the weight recovery rate and recovery time.

The detection ratios and response times obtained with 1-cycle and 5-cycle recovery time are shown in Figure 6. The weight recovery rate was chosen from 0.02 to 1. For the 1-cycle recovery time case, larger weight recovery rate shows slightly better performance as a little higher detection and lower misdetection ratios obtained. However, a long response time is required when large recovery rate is used. In the case of 5-cycle recovery time, the similar trend is observed. There is no big difference in the detection rate and response time between the 1-cycle and 5-cycle cases, but the average misdetection ratio using 5-cycle recovery time is approximately 10% higher than that using 1-cycle.

**Figure 6** Detection accuracies (top) and response time (bottom) under different weight recovery rates (see online version for colours)

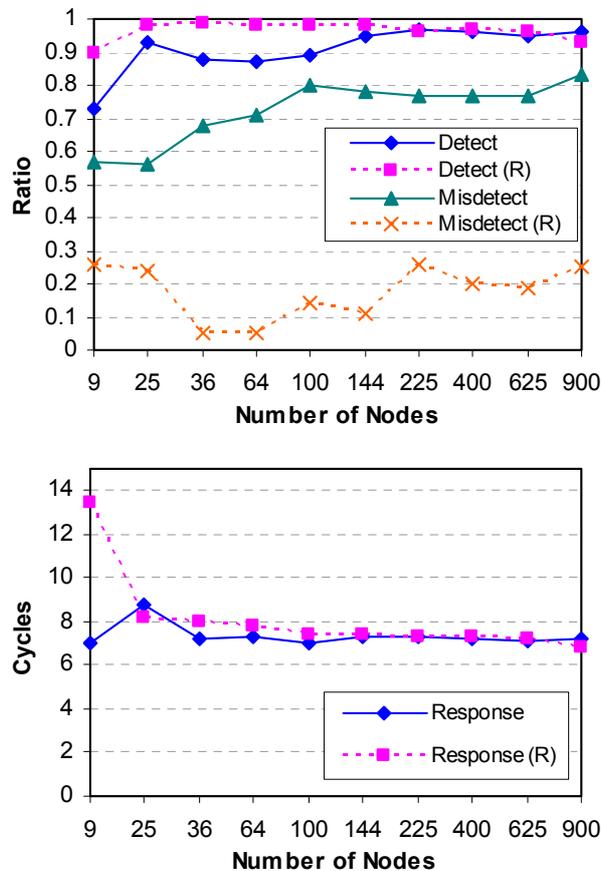


These results imply that large weight recovery rate and short recovery time can achieve overall good performance for the weight recovery mechanism. Note that this simulation is based on the assumption that the malicious node keeps reporting wrong information once compromised, thus the weights of malicious nodes are not merely increased by the recovery mechanism. If malicious nodes send wrong information intermittently, it is empirically shown that a moderate recovery rate and long recovery time can be used to avoid weight recovery on malicious nodes. For this reason, in the following experiments we chosen an intermediate value (0.1) as the weight recovery rate and 1 cycle as the recovery time.

#### 4.4 Scalability

We further evaluated the scalability of the algorithm with various numbers of nodes. Figure 7 shows the results of the algorithm varying the number of nodes from 9 to 900. The results of the algorithm without weight recovery mechanism are also presented for comparison. Other parameters including the attack and alarm probabilities are the same as those used in the experiments of Section 4.2.

**Figure 7** Detection accuracies (top) and response time (bottom) with various numbers of sensor nodes (see online version for colours)



Note: (R) indicates the algorithm with the weight value recovery mechanism.

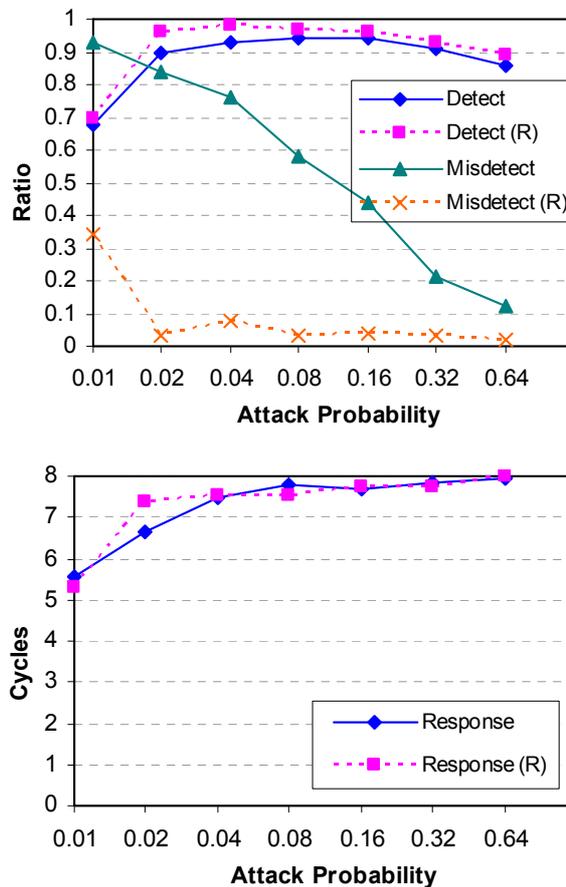
The response time, detection, and misdetection ratios are stable while the number of nodes are increased from 9 to 900, particularly when the number of nodes is greater than 100. These results indicate that the WTE based detection algorithm has good scalability as it works well under variant network sizes without large performance degradation. Especially if the size of network becomes large enough, For instance, greater than 225, the network size almost has no influence on the performance.

Compared with the one without the weight recovery mechanism employed, the algorithm incorporating the recovery mechanism has overall higher detection ratio and much lower misdetection ratio, with about 0.5 down in the misdetection ratio. These results well demonstrate the effectiveness of the recovery mechanism.

#### 4.5 Robustness

Finally, the performance at various attack probabilities was evaluated for 100 nodes case. Similarly, the two cases: with and without the weight recovery mechanism, are both investigated. Figure 8 shows the evaluation results including the response time, detection, and misdetection ratios. As mentioned previously, the detection is terminated when more than 25% of the total nodes are detected as malicious nodes. The attack probability actually is the ratio of the malicious node out of 25% of the total number of nodes.

**Figure 8** Detection accuracies (top) and response time (bottom) with various different attack probabilities (see online version for colours)



Note: (R) indicates the algorithm with the weight value recovery mechanism.

The increasing attack probability means that there are more nodes being compromised and falsified data are inserted. While there are only small changes observed in detection ratios, the misdetection ratio decreases largely as the growth of the attack probability. This is partially due to the detection become relatively easy, or the misdetection hardly occurs when a large number of malicious nodes existing. The response time slightly increases with attack probability increasing because as more malicious nodes appear, the aggregated data is affected more by the falsified data.

Similar to what reported in the previous section, the algorithm with the recovery mechanism incorporated shows higher detection ratio and significant lower misdetection ratio, verifying again that the recovery mechanism is able to largely improve the overall performance of the algorithm.

As the results reported above, although the performance of the detection algorithm is largely depending on the setting of parameter values, the response time, detection, and misdetection ratios are promising with optimal settings, especially in the cases a large number (for instance, more than 100) of nodes deployed in the group under the same FNs. The results also demonstrate that the proposed WET based detection algorithm is effective for both large networks and high attack probability conditions. Moreover, the recovery mechanism could largely reduce the misdetection ratio without any negative effects on the detection ratio and response time.

## **5 Conclusions**

In this paper, we proposed a novel WTE based algorithm for the detection of malicious SNs in WSNs. The basic idea is that a weight representing the reliability of a node is assigned to each SN in the cluster under a FN. Since malicious nodes usually report falsified information to disrupt the network, if a node sends incorrect information, the FN gradually decreases the weight of the node and detect the node as a malicious node when its weight value becomes lower than a threshold. In addition, a weight recovery mechanism is incorporated in the algorithm to recover the weight of a node whose weight is accidentally decreased.

The simulation experimental results have shown that the WTE algorithm is a promising solution to address the malicious nodes detection problem in WSNs. It achieves good scalability with reasonable detection delay, and is applicable to variant numbers of SNs deployed under the control of a FN, thus suitable to a flexible node deployment in WSNs. Note that the size of a cluster under a FN could be adjusted by setting more and less FNs for a WSN with fixed size. Essentially, it could be treated as a node-clustering problem. In the robustness simulation regarding attack probability, the algorithm also shows effectiveness when the network is exposed to a heavy attack conditions. In both scalability and robustness simulations, the misdetection ratios in these cases could be largely reduced by introducing the weight recover mechanism.

Although there are several other research works addressing the malicious node detection problem in WSNs reported, it is difficult to compare the performance between each other. As introduced in Section 2, the design assumptions and the experiment environments are very different. Particularly, lack of a widely recognised benchmark makes it meaningless to compare the results, for example, the definition of detection ratio.

The proposed algorithm is based on the assumption that base stations (FNs and APs) are points of trust. In practice, if the adversary can gain control over the base stations, it can launch any possible attacks against the WSN. Although this assumption is an interesting issue needed to be further discussed, it is beyond the scope of this paper. Another critical assumption is that the majority of the SNs are working properly. If the number of compromised nodes exceeds the number of normal nodes, normal nodes could be reported as malicious ones and malicious nodes are treated nice ones.

In this paper we reported merely some preliminary results, which verified the correctness and effectiveness of our solution. More detailed analysis regarding the performance of our algorithm needs to be studied in the ongoing research and more questions to be answered. In our progressive efforts, we are studying the deployment of FNs and the influence of different densities of FNs on the performance. In addition, apart from the simulation, we are setting up a physical testbed consisting of more than 64 SNs. 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.

## References

- Ayday, E., Delgosa, F. and Fekri, F. (2007) 'Location-aware security services for wireless sensor networks using network coding', *Proceeding of the 26th Annual IEEE Conference on Computer Communications (IEEE INFOCOM 2007)*, Anchorage, Alaska, USA, 6–12 May.
- Chen, R., Park, J.M. and Bian, K. (2006) 'Robust distributed spectrum sensing in cognitive radio networks', Technical Report TR-ECE-06-07, Department of Electrical and Computer Engineering, Virginia Tech., July.
- Curiac, D-I., Baniias, O., Dragan, F., Volosencu, C. and Dranga, O. (2007) 'Malicious node detection in wireless sensor networks using an autoregression technique', *The 3rd International Conference on Networking and Services (ICNS'07)*, Athens, Greece, 19–25 June.
- Du, W., Fang, L. and Ning, P. (2005) 'LAD: Localization Anomaly Detection for wireless sensor networks', *The 19th International Parallel and Distributed Processing Symposium (IPDPS'05)*, Denver, Colorado, USA, 3–8 April.
- Estrin, D., Govindan, R., Heidemann, J. and Kumar, S. (1999) 'Next century challenges: scalable coordination in sensor networks', *MOBICOM*, August.
- Ganeriwai, S. and Srivastava, M.B. (2004) 'Reputation-based framework for high integrity sensor networks', *SASN'04*, Washington, DC, USA, 25 October.
- Hu, Y., Perrig, A. and Johnson, D. (2003) 'Packet leashes: a defense against wormhole attacks in wireless ad hoc networks', *Proceeding of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM 2003)*, San Francisco, California, USA, 30 March–3 April.
- Junior, W., Figueriredo, T., Wong, H-C. and Loureiro, A. (2004) 'Malicious node detection in wireless sensor networks', *The 18th International Parallel and Distributed Processing Symposium (IPDPS'04)*, Santa Fe, New Mexico, USA, 26–30 April.
- Karlof, C., Sastry, N. and Wagner, D. (2004) 'TinySec: a link layer security architecture for wireless sensor networks', *ACM Sensys*, November.
- Karlof, C. and Wagner, D. (2003) 'Secure routing in wireless sensor networks: attacks and countermeasures', *Journal of Ad Hoc Networks*, Elsevier.
- Lamport, L., Shostak, R. and Pease, M. (1982) 'The Byzantine generals problem', *ACM Transactions on Programming Languages and Systems*, July, Vol. 4, No. 3.
- Luo, H., Zerfos, P., Kong, J., Lu, S. and Zhang, L. (2002) 'Self-securing ad hoc wireless networks', *IEEE ISCC (IEEE Symposium on Computers and Communications)*, Italy.

- Newsome, J., Shi, E., Song, D. and Perrig, A. (2004) 'The Sybil attack in sensor networks: analysis and defense', *International Symposium on Information Processing in Sensor Networks*, Vol. 1.
- Przydatek, B., Song, D. and Perrig, A. (2003) 'SIA: Secure Information Aggregation in sensor networks', *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, Los Angeles, California, USA, 5–7 November.
- Servetto, S.D. (2006) 'From "small sensor networks" to "sensor networks"', *EmNets 2006*, May.
- Sun, B., Wu, K. and Pooch, U. (2002) 'Secure routing against black-hole attack in mobile ad hoc networks', *Proceedings of the 2002 Conference on Communications and Computer Networks (CCN 2002)*, Cambridge, Massachusetts, USA, 4–6 November.
- Sun, Y., Han, Z., Yu, W. and Liu, K.J.R. (2006) 'A trust evaluation framework in distributed networks: vulnerability analysis and defense against attacks', *Proceeding of the 25th Annual IEEE Conference on Computer Communications (IEEE INFOCOM 2006)*, Barcelona, Catalunya, Spain, 23–29 April.
- Tubaishat, M. and Madria, S. (2003) 'Sensor networks: an overview', *IEEE Potentials*, April, Vol. 22, No. 2, pp.20–23.
- Vieira, M.A.M., da Silva, D.C., Jr., Coelho, C.N., Jr. and da Mata, J.M. (2003) 'Survey on wireless sensor network devices', *Emerging Technologies and Factory Automation (ETFA03)*, September.
- Yu, Y., Krishnamachari, B. and Prasanna, V.K. (2004) 'Energy-latency tradeoffs for data gathering in wireless sensor networks', *Proceeding of the 23rd Conference of the IEEE Communications Society (IEEE INFOCOM 2004)*, Hong Kong, China, 7–11 March.
- Zhao, S., Tepe, K., Seskar, I. and Raychaudhuri, D. (2003) 'Routing protocols for self-organizing hierarchical ad-hoc wireless networks', *Proceedings of the IEEE Sarnoff Symposium*, Trenton, New Jersey, March.
- Zhou, L. and Haas, Z. (1999) 'Securing ad hoc networks', *IEEE Network Special Issue on Network Security*, November, Vol. 13, No. 6, pp.24–30.
- Zhu, S., Setia, S. and Jajodia, S. (2003) 'LEAP: efficient security mechanisms for large-scale distributed sensor networks', *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS 2003)*, Washington, DC, USA, 27–30 October.