WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2010; **10**:559–572
Published online 15 April 2009 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/wcm.785

A lightweight intrusion detection framework for wireless sensor networks

Tran Hoang Hai[1], Eui-Nam Huh[1*,†] and Minho Jo[2**,‡]

[1]*Internet Computing and Network Security Laboratory, Kyunghee University, Seocheon-Dong, Giheung-Gu, Yongin-Si, Gyeonggi-Do, South Korea*
[2]*Graduate School of Information Management and Security, Korea University, 1, 5-Ga, Anam-Dong, Sungbuk-Gu, Seoul 136-701, South Korea*

Summary

In recent years, Wireless Sensor Networks (WSNs) have demonstrated successful applications for both civil and military tasks. However, sensor networks are susceptible to multiple types of attacks because they are randomly deployed in open and unprotected environments. It is necessary to utilize effective mechanisms to protect sensor networks against multiple types of attacks on routing protocols. In this paper, we propose a lightweight intrusion detection framework integrated for clustered sensor networks. Furthermore, we provide algorithms to minimize the triggered intrusion modules in clustered WSNs by using an over-hearing mechanism to reduce the sending alert packets. Our scheme can prevent most routing attacks on sensor networks. In in-depth simulation, the proposed scheme shows less energy consumption in intrusion detection than other schemes. Copyright © 2009 John Wiley & Sons, Ltd.

KEY WORDS: intrusion detection; wireless sensor networks (WSNs); routing attacks; lightweight; security

1. Introduction

Wireless Sensor Networks (WSNs) have grown to become one of the most promising and interesting fields over the past few years. WSNs are wireless networks consisting of distributed sensor nodes that cooperatively monitor physical or environmental conditions. A sensor node is a tiny and simple device with limited computational resources. Sensor nodes are randomly and densely deployed in a sensed environment. WSN is designed to detect events or phenomena, and collect and return sensed data to the user.

WSNs have been used in many applications such as battlefield surveillance, traffic monitoring, healthcare, environment monitoring, etc. Some basic features of sensor networks are as follows [1]:

- Self-organization
- Short-range broadcast communication and multi-hop routing

**Correspondence to: Eui-Nam Huh, Department of Computer Engineering, Kyung Hee University, Seocheon-Dong, Giheung-Gu, Yongin-Si, Gyeonggi-Do 446-701, South Korea.
†E-mail: johnhuh@icns.khu.ac.kr
**Correspondence to: Minho Jo, Graduate School of Information Management and Security, Korea University, 1, 5-Ga, Anam-Dong, Sungbuk-Gu, Seoul 136-701, South Korea.
‡E-mail: minhojo@korea.ac.kr

- Dense deployment and cooperative sensors
- Frequently changing topology due to fading and node failures
- Limitations in computational resources such as energy and memory.

The characteristics of wireless infrastructure and characteristics of WSNs cause potential risks of attacks on the network. Numerous studies have attempted to address vulnerabilities in WSNs such as denial of service in sensor networks [2] and secure routing in sensor networks [3]. Current research on security in sensor networks generally focuses on secure routing protocols, key management, and prevention techniques for specific attacks [4–6]. Although research on security (related to) issues in WSN is productive, the need for a security framework for WSNs still exists.

Intrusion detection system (IDS) is a common prevention mechanism that protects the network from intrusion. In this paper, we study the problem of intrusion detection in WSNs and propose a hybrid intrusion detection framework for clustered sensor networks. Our scheme suits the demands and restrictions of the infrastructure and characteristics of WSNs. The analytical analysis and simulation result show that our IDS scheme can detect more than 90% of malicious nodes under various attacks, with a high rate of packet collision. The contribution of our paper is as follows:

- A distributed IDS framework for creating, updating, and evaluating alert packets in clustered WSNs.
- Detection of common routing problems and attacks in clustered WSNs, based on neighbor knowledge and routing rules.
- Use of a reputation system as the basis of self-triggering IDS modules and evaluation of the alert packet from monitor nodes.
- Reduction of alerts using over-hearing to reduce energy consumption in IDS modules.
- High detection rate under burst attacks.

The paper is further organized as follows: In Section 2, we review and study the problem of application of IDS in WSNs and outline the challenges. Section 3 proposes our security architecture and detection algorithms for WSNs. In Section 4, we provide two algorithms to self-trigger and reduce energy consumption in IDS modules. Section 5 provides the simulation and performance analysis. Finally, the paper ends with a conclusion and future work.

2. Security in WSNs

2.1. Routing Threats

The design of routing protocols in sensor networks never considers security as a primary goal. Routing protocols in sensor networks are simpler and more susceptible to attacks than the other two types of wireless networks: ad hoc and cellular.

The first serious discussion and analyses on secure routing were performed by Karlof and Wagner in 2003 [3]. They studied multiple types of attacks on routing protocols in detail and the effects on common routing protocols in WSNs. The assumption is that there are two types of attacks, outside attacks and inside attacks. In this paper, we only examine inside attacks. Outside attacks are prevented by using link layer security mechanisms [7]. They propose two types of adversaries, a mote-class adversary and laptop-class adversary. In the mote-class adversary, the adversary accesses a few sensor nodes with capabilities similar to legitimate nodes. These nodes are tampered with and reprogrammed for an adversary's purpose. In the laptop-class adversary, the adversary accesses more powerful devices such as a laptop with greater battery power, high CPU processing rate and high-power radio transmitter. In this case, the adversary has more opportunities to deploy attacks on the network. In this section, we review the most common network layer attacks on WSNs and highlight the characteristics of these attacks [3].

2.1.1. Selective forwarding

In a selective forwarding attack, malicious nodes prevent the flow of routing information in sensor networks by refusing to forward or drop the messages traversing them [3]. Another aspect of this type of attack is that malicious nodes may forward the messages along an incorrect path, creating inaccurate routing information in the network.

2.1.2. Sinkhole

In a sinkhole attack, the adversary redirects nearly all the traffic from a particular area via a malicious node, creating a metaphorical sinkhole [3]. The laptop-class adversary may use higher computational resources and communication power, than a legitimate node, to advertise itself as the shortest path to the base-station or, in our case, the cluster head (CH). A CH aggregates

DOI: 10.1002/wcm

the data of member nodes in a cluster and relays them to another CH or the sink node.

2.1.3. Wormhole

In a wormhole attack, the adversary tunnels messages received in one malicious node and replays them in a different part of the network. The two malicious nodes usually claim that they are merely two hops from the base station. Khalil suggests five modes of wormhole attacks in his paper. Details of these modes are in References [8,9].

2.1.4. Hello flood attack

Many routing protocols use "hello" broadcast messages to announce themselves to their neighbor nodes. The nodes that receive hello messages assume that source nodes are within range and add source nodes to their neighbor list. The laptop-class adversary can spoof hello messages with sufficient transmission power to convince a group of nodes that they are its neighbor.

2.1.5. Sybil attack

In this attack, a malicious node can present multiple identities to other nodes in the network. The Sybil attack poses a significant threat to most geographic routing protocols. Sybil attacks are prevented via link layer authentication [10]. Within the limited scope of this paper, we assume that the Sybil attack is prevented via authentication; so the combination of Sybil with other attacks is not considered in this paper.

2.2. Intrusion Detection System in Wireless Networks

IDS is defined as a system that tries to detect and alert of attempted intrusions into a system or a network [11]. IDSs are classified into two major approaches: misuse detection and anomaly detection. Each approach has its own unique advantage. The misuse technique has the advantage that it can detect most known attacks in a rule database. But, new attacks require new rules to be constructed and distributed [12,13]. The anomaly technique has the advantage that it does not require any rules and can detect novel attacks. The main disadvantage of anomaly detection is the high false positive rate [14–16]. Although IDS is used as a major prevention mechanism in wired networks, it is difficult to apply IDS in wireless networks, because of the vast difference in network characteristics.

Sensor networks inherit all aspects of wireless networks and they have their own distinct characteristics that make the design of a security model for sensor networks different from that of ad hoc networks. The batteries in sensor networks may not be rechargeable; thus, we cannot recharge or replace the batteries if sensor nodes use excessive computational resources to process the data.

Sensor networks are constrained in resource compared to ad hoc and cellular networks [17]. A typical sensor node such as MICA has an 8 MHz microprocessor, 128 Kb program flash memories, and 512 Kb serial flash memories [18]. WSNs are deployed more densely and randomly in the environment and sensor node failure is likely to happen. So, it is impossible for a sensor node to store the signature data about malicious nodes for the whole network in a manner similar to additional misuse detection. Also, it is very difficult to use traditional anomaly detection methods in WSNs because sensor nodes cannot monitor all the traffic traversing them and compute anomalous events. These specific characteristics of WSN demand a novel design of the security architecture for such an environment. Although wireless ad hoc networks and WSNs share some common characteristics, and there was development of IDS in a wireless ad hoc network [19], Roman showed in his paper that they can not be directly applied in WSNs [20]. They proposed a novel technique for optimal monitoring of neighbors called "spontaneous watchdog," which extends the "watchdog monitoring mechanism" in Reference [21]. The problem with this approach is that the author fails to consider the selection of a global agent. Another weakness of this approach is that it does not deal with the collision of packets, which is likely due to the high density of nodes in WSNs. Onat *et al*. (2005) proposed an anomaly detection based on security scheme for WSNs. In their method, each sensor node builds a simple statistical model of its neighbor's behavior, and these statistics are used to detect changes [22]. The system features which analyze anomalies are the average of received power and packet arrival rate. Their system cannot detect selective forwarding and wormhole attacks because of their simple statistical features. Banerjee *et al*. (2005) proposed an intrusion detection mechanism based on an ant colonies system [23]. Their basic idea is to identify the affected path of intrusion in the sensor network, by investigating the pheromone concentration. However, they do not specify the detailed solution to routing attacks. In 2006, Techateerawat *et al*. published a paper in which they

DOI: 10.1002/wcm

designed an intrusion framework based on the layout and selection of monitor nodes [24]. They proposed a voting algorithm for selection of nodes which must trigger their IDS agent. Their approach not only reduced the monitor nodes and energy consumption in networks but also reduced the probability of detection. Unfortunately, their detection algorithms were not demonstrated in detail. A recent study of Chong *et al*. (2006) developed an intrusion detection scheme that uses a clustering algorithm to build a model of normal traffic behavior. Then, they used this model to detect anomalous traffic patterns [25]. Silva *et al*. proposed a decentralized IDS scheme, based on the specification in [26]. In these two schemes, every IDS agent functions independently, and can detect signs of intrusion locally, by observing all data received, without collaboration between its neighbors. They tried to apply an anomaly technique based on wired networks for WSNs, so their scheme incurs excessive computational resource consumption in each node. Agah *et al*. applied game theory in order to build a detection framework for denial of service in WSNs. However, their scheme is not specified for routing attacks in WSNs [27]. There are multiple IDS proposals for WSNs, but many are incomplete or only focus on a specific attack [28]. Our contribution in this paper is based on previous work and involves the creation of a novel, efficient IDSs for WSNs. Futhermore, we propose a simple selection algorithm to trigger IDS modules in particular nodes. Our algorithm minimizes the monitor nodes that must trigger the intrusion detection modules, thus enhancing the network lifetime.

3. A Lightweight Intrusion Detection Framework for Sensor Networks

3.1. Architecture

In sensor networks, multiple routing protocols, power management, and data dissemination are designed in which energy and computational resources are essential designs. Cluster-based routing protocols were developed for sensor networks (LEACH, HEED, PEGASIS, TEEN, and APTEEN [29]) to achieve scalability, power savings, data routing redundancy, etc. Routing is usually separated into two phases: the setup phase and the steady phase. In the setup phase, the cluster is organized and CHs are randomly selected and rotated to distribute the energy load among the network. In the steady phase, the CHs receive all data in their clusters and send aggregated data to the base station, to reduce the amount of information arriving at the base station. In our IDS architecture, every node belongs to a single cluster among the clusters that are geographically distributed across the whole network. Our aim is to utilize cluster-based protocols in energy saving, reduced computational resources and data transmission redundancy. In this section, we propose an intrusion framework for information sharing, which utilizes hierarchical architecture to improve intrusion detection capability for all participating nodes. Previous work on the application of IDS for sensor networks was undertaken by Roman [20]. The author suggested general guidelines for the application of IDS to WSNs, which influenced our work. In addition, our proposed intrusion detection framework is influenced and improved by previous works in [8,26,30].

In our scheme, an IDS agent is located in every sensor node. Each sensor node has two intrusion modules, called local IDS agent and global IDS agent. Because of the limited battery life and resources, each agent is only active when it is needed.

3.1.1. Local agent

The local agent module is responsible for monitoring the information sent and received by the sensor. The node stores an internal database, named a blacklist, about specific malicious nodes in network. When the network is initially configured, the sensor nodes lack any knowledge about malicious nodes. After the deployment of WSNs, the signature database is gradually constructed. The entry into the malicious node database is created and propagated to every node by CHs.

3.1.2. Global agent

The global agent is responsible for monitoring the communication of its neighbor nodes. Because of the broadcast nature of wireless networks, every node can receive all packets within its communication range. We use the watchdog monitoring mechanism and pre-defined routing rules with two-hop neighbor knowledge to monitor these packets. If the monitor nodes discover a potential breach of security in their radio range, they create and send an alert to the CHs. Then, the CHs receive the alert and make the decision about a suspicious node. Both agents are implemented in the application layer.

DOI: 10.1002/wcm

3.2. Detection Algorithms

We assume that when a sensor node is first deployed in the environmental field, an adversary requires a particular period of time to deploy an attack. This implies that no malicious node appears during the initial stage of sensor node deployment.

The monitor nodes use the watchdog monitoring mechanism and predefined rules with two-hop neighbor knowledge to detect anomalies within their transmission ranges as shown in Figure 1. In watchdog, due to the broadcast nature of wireless networks, monitor nodes receive packets within their radio range. These packets are captured and stored in a buffer that contains information including the packet identification and type, source and destination, etc. Each entry in the buffer is time stamped. This expires after a timeout or after the entry in the buffer is examined by monitor nodes.

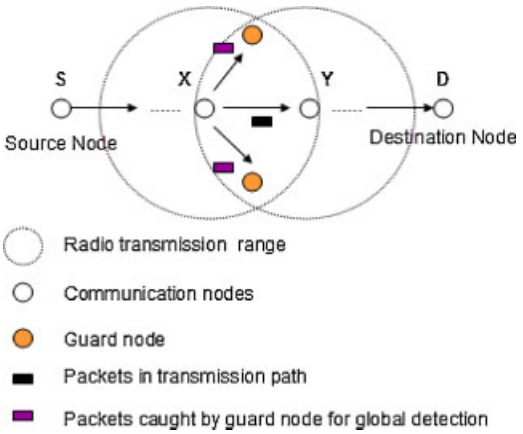

Fig. 1. Monitor node.

3.2.1. Data structure

Sensor nodes maintain two databases: malicious nodes and neighbor knowledge.

3.2.1.1. Two-hop neighbor knowledge. Two-hop neighbor knowledge is generally used in broadcasting protocols to reduce the number of transmissions such as Source-based Protocol, Dominant Pruning, etc [31]. As we mentioned in related work, Khalil *et al*. [8] applied two-hop neighbor knowledge to detect wormhole attacks in WSNs and ad hoc networks.[8,9] We also apply two-hop neighbor knowledge as a component of our detection technique. Unlike the two-phase setup in Khalil's work, we establish our two-hop neighbor list in each sensor node via a single phase, by modifying the hello packet. When the sensor nodes are initially deployed in the sensing environment, each node must build its direct neighbor list and a list of two-hop neighbors accessible to these one-hop neighbors. To accomplish this, each node broadcasts its hello message; fields contain information about source node ID, immediate node, and the hop counter is set to two. In the case of the source node, the source node ID and immediate node have the same node ID (). When a node receives a two-hop hello packet, it changes the immediate node as its node ID, decrements the hop count to one and re-broadcasts it. The sensor node receiving this hello message assigns the immediate node as its direct neighbors and the source node as its two-hop neighbor. This process is performed once after the deployment of sensor nodes. We make the assumption that the neighbor node knowledge is secure and confidential within the deployment period.

3.2.2. Malicious node database/blacklist

This internal database is computed and generated in the CH via the use of anomaly detection in the global detection algorithms of monitor nodes. Once a monitor node discovers an anomalous event within its neighborhood, it creates and sends an alert to its CH. If the malicious counter from a suspicious node stored in a CH crosses a threshold X, the CHs create and propagate a new rule to every sensor node in the cluster. The sensor nodes update the new rule and add the entry to its malicious database. The malicious node is isolated from the cluster and not involved in communication in the network. CH serves as an intrusion data collection point. The rule must contain the following fields: time of creation, classification (type of alert), and source of the alert [32].

3.2.3. Pre-defined routing rules

When the sensor node is initially deployed, there is no entry in its internal malicious node database, except for some predefined, simple rules in the global agent. The global agent uses pre-defined rules and the two-hop neighbors' list to monitor communication in their neighborhood. These rules help monitor nodes detect common problems and specific attacks on routing protocols, based on a previous work [24]. In our scheme, these rules are adapted to the routing protocols used.

DOI: 10.1002/wcm

- *Interval rule*: An alert is created by monitor nodes if the period between the receptions of two consecutive packets exceeds the allowed limit.
- *Integrity rule:* The packet payload must be the same along the path on a transmission link.
- *Delay rule*: The delay of a packet from one node must be limited to the timeout period.
- *Radio transmission range rule*: All packets received by a monitor node must originate from among its neighbors or a previous hop; via the estimation of the average receive power (dBm).
- *Neighbor rule*:
  (1) The monitor node waits to determine if the destination node forwards the packet along the path to the sink. If not, it sends an alert packet to the CH.
  (2) The monitor node waits to detect the packet that was forwarded along the path to the sink. It checks its two-hop neighbor knowledge to determine if the destination node of the forwarded packet is on the right path to the sink. If not, it sends an alert packet to the CHs.

When a sensor node receives a packet from a sensor in the network, if the source node's ID is in its black list then the sensor node uses Local_function() to drop the packet. If both source and destination's node are its one-hop neighbors, it triggers the Global_detection function. The algorithm is illustrated in Figure 2. The global detection modules use two-hop neighbor knowledge and routing rules to detect anomalies within their transmission ranges. The illustration of Gobal_function() is represented in Figure 3.

The CHs are responsible for alert aggregation from monitor nodes and computation. If the number of alerts about a suspicious node crosses the threshold *X*, the CHs create a rule and propagate it to every node in the cluster. The algorithm is illustrated in Figure 4.

```
Communication Node
1.  Repeat <listen to the packet>
2.   Check <packet header>
3.   If {ID = destination node's ID} {
4.        If Local_Detection(packet)
5.            Then drop(packet)
6.        Else receive(packet);
7.       }
8.   And If (source & destination's ID, 1
hop neighbor)
9.           Then Global detection (packet)
10.       Else Drop (packet)
11. Until No transmission
```

Fig. 2. Algorithm of activating monitor nodes.

```
Global_detection(packet_i)
1. {
2.  If Looking(packet_id, buffer)
3.   then {
4.     If Check(node's ID, 2 hop neighbor's
5.         list )
6.      Or Check(packet_i, predefined-rules)
7.     then {
8.             Create(alert);
9.             Send(alert, cluster_head);
10.    }
11. }
```

Fig. 3. Global detection at monitor nodes.

```
Cluster head
1. Repeat
2.   If Looking (alert, intrusion alert)
3.   Then {
4.         Malicious count (node) ++
5.         If (Malicious count (node) > X)
6.         Then {
7.               Create (rule);
8.               Propagate (rule);
9.               }
10.       }
11. Until No transmission
```

Fig. 4. Alert computation at the CH.

By applying our proposed algorithm, the following attacks introduced in Section 2 are detected easily.

3.2.4. Detection of selective forwarding

In the selective forwarding attacks, the transmission link from node A to node B is monitored by their monitor nodes. For instance, nodes X, Y, and Z catch and store the packets going out of node A with node B as their next intermediate node. If node B tries to stop or drop these packets, the monitor nodes will create and send an alert to CH. The monitor nodes can also use the predefined rules to check if node B forwards the packet in the right path. If node B tries to send the packets to wrong path by forwarding to an unknown node, the monitor nodes will check their two-hop neighbor nodes' list. If the destination node's identification of the forwarded packet is not in node B's neighbor list, the monitor nodes will send an alert to CH. After the packets are fowarded to the right path, the entry in the monitor node's intrusion buffer will be removed.

3.2.5. Detection of sinkhole and hello flood

The common feature between the two attacks is that the malicious node will convince it as the nearest path to the base station by using high power transmission. All packets going to node A must be originated from node A's neighbor list. The monitor nodes use the neighbor's

DOI: 10.1002/wcm

list and the predefined signal rule to check if a packet is originated from a far located node.

3.2.6. Detection of wormhole

Our systems can detect four types of wormhole attacks by inheriting the advantage of local monitoring mechanism. We use the two-hop neighbors' list and the predefined rules to improve the detection of wormhole in clustered WSNs.

4. Optimal Triggering of Intrusion Detection Modules

In our scheme and previous work, every node participates in the intrusion detection, so the network lifetime is potentially quickly reduced, because the workload is concentrated in IDS modules. In this section, we provide two algorithms to reduce the energy consumption in IDS modules in WSNs. Current research on intrusion detection and prevention techniques in WSNs are generally built on the assumption of a trusted environment. Unfortunately, sensor nodes are randomly deployed in an unknown, hostile environment, so they cannot be trusted. A disadvantage of cooperative IDS is the detection accuracy of IDSs because they cannot evaluate alerts from monitor nodes. By using a lightweight trust-based framework as the basis of cooperative IDSs, we can overcome this problem and evaluate alerts from monitor nodes based on their trust values. Evaluation of alerts arriving at CHs makes our IDS scheme more resilient and accurate. We can apply any reputation framework for WSN as an integrated part in our IDS scheme.

4.1. Triggering Based on Trust Priority

Trust is defined as the level of trustworthiness of a particular node. $T\mathrm{v}_{\mathrm{xy}}$ is the trust value of node Y calculated by node X. In our schemes, we require each sensor node to maintain a reputation table of its neighbors; the reputation value is a metric of trust. A reputation table is a small database of trust values of direct neighbor nodes as, for example, node X.

$$T\mathrm{v}_{\mathrm{X}} = (T\mathrm{v}_{\mathrm{X},1}, T\mathrm{v}_{\mathrm{X},2}, \ldots, T\mathrm{v}_{\mathrm{X,N}}) \tag{1}$$

where $T\mathrm{v}_{X,i}$ represent the trust value of the ith neighbor node of X. Calculation and update of reputation tables in sensor nodes can be found in Reference [33]. Our reputation system is fully adaptive with detection modules, because both schemes are based on an overhearing mechanism. Each sensor node calculates the average trust of its neighbor nodes with the following equation:

$$E[\mathrm{X}] = \frac{\sum_{i=1}^{N} T\mathrm{v}_{\mathrm{X},i}}{N} \tag{2}$$

where E[X] represents the average trust value of X's neighbor nodes. The trust value is classified by the following mapping function:

$$M\mathrm{p}\,(T\mathrm{v}_{\mathrm{node}}) = \left\{ \begin{array}{l} \mathrm{high} - 0.8 \leq T\mathrm{v}_{\mathrm{node}} \leq 1 \\ \mathrm{medium} - 0.5 \leq T\mathrm{v}_{\mathrm{node}} \leq 0.8 \\ \mathrm{uncertain} - 0.3 \leq T\mathrm{v}_{\mathrm{node}} < 0.5 \\ \mathrm{low} - 0 \leq T\mathrm{v}_{\mathrm{node}} < 0.3 \end{array} \right\} \tag{3}$$

After calculating the trust average, the sensor node sets this value according to the mapping function above to indicate the trust level requirement. Only nodes having a better than average trust value can trigger the global agent for cooperative detection. Each packet includes its own trust requirement (high, medium, or uncertain) in its header. Thus, only sensor nodes with a trust value better than the trust requirement can trigger their global agent. However, if a sensor node with a low trust value tries to send a false alert packet to the CHs, the CHs drop the alert packet, and its trust value is reduced for its malicious behavior. In our case, nodes having a low trust value cannot trigger or participate in the intrusion detection.

4.2. Evaluation of Alert Packets

The CHs are responsible for alert aggregation and computation. We propose four levels of trust, so we can compute the alert counter in each malicious node, based on trust states of our monitor nodes. The malicious counter is defined as the threshold of malicious activities of a sensor node that cannot be exceeded. If the malicious counter of a sensor node exceeds the threshold, the sensor node is revoked from the cluster and WSNs. We suggest four parameters $(\lambda, \beta, \delta, \varphi)$ associated with four trust levels of a monitor node's incoming alert packet, in our proposed scheme$\lambda = 0$. The equation for computing the alert counter of a

DOI: 10.1002/wcm

```
Cluster head
1.  Repeat
2.    If Looking (alert, intrusion alert)
then {
3.      Case Trust level node of
4.           'High': MC = MC + λ;
5.           'Medium': MC = MC + β;
6.           'Uncertain': MC = MC + δ;
7.      End Case
8.  If (Malicious count (node) > X) then {
10.          Create (rule);
11.         Propagate (rule);
12.          }
13  }
14. Until No transmission
```

Fig. 5. Improved alert computation algorithm at the CHs.

malicious node is described as follows:

$$\mathrm{MC}_{\mathrm{node}} = \beta \sum_{j=1}^{i} i + \delta \sum_{k=1}^{k} j + \varphi \sum_{l=1}^{l} k \qquad (4)$$

where $0 < \beta < \delta < \varphi < 1$and i, j, and k are the number of alert packets with the correlative trust states mentioned above. So, the aggregation and computation of alert packets at CHs is improved as Figure 5 below. By setting the trust-requirement as the average of the trust, we can reduce participation of sensor nodes in the intrusion detection, while providing high trustworthiness of incoming alert packets.

By setting the trust-requirement as the average of the trust, we can reduce participation of sensor nodes in the intrusion detection, while providing high trustworthiness of incoming alert packets.

4.3. Reduction of Alert Packets Using Over-Hearing

In some cases of deployment, there are multiple sensor nodes concentrated in a small area. Consequently, if there is malicious activity in a link, multiple alert packets may be transmitted to CHs from different monitor nodes in an instant. The major issue in this case is the redundancy of the transmission of alert packets to CHs, which can cause collisions and waste energy on transmission of the same alert packets. Until now, in a given case, we need a single alert packet sent simultaneously to CHs, for malicious activity. If a single alert packet is sent at the instant malicious activity occurs, we can reduce redundant alert packets, thus reducing energy consumption in monitor nodes. To resolve this problem, we apply an over-hearing mechanism for the medium access control (MAC) layer. Over-hearing is not a new approach. It was initially applied in 802.11 [34], where nodes use over-hearing to determine when the channel is free. In Reference [33], the authors extended S-MAC to event-driven applications, where there are multiple redundant transmissions. The principle of our approach is very simple. When malicious activity occurs in a transmission link, multiple monitor nodes are aware of this malicious activity, and prepare alert packets to send to the CHs. If a monitor node does not obtain the medium to send an alert packet, it knows there is a transmission within range. The monitor node buffers the alert packet and over-hears the packets sent within range. If the monitor node detects a neighbor sending the same alert packet, it drops the alert packet in its buffer. Otherwise, the monitor node sends the alert packet until it obtains the medium. Using this method, we can reduce both the number of transmissions and the number of collisions in sending the same alert packets of monitor nodes. The study in [34,35] found that each bit transmitted in WSNs consumes power about equivalent to executing 800–1000 instructions. Thus, we can minimize the power consumption in detection modules, because communication is more costly than computation in WSNs.

5. Performance Analysis

In this section, we analyze and evaluate the proposed detection capability, to determine the performance of our schemes. The probability of detection of an attack P_{D} depends on the following three factors: number of monitor nodes, probability of a missed detection of a monitor node, and our malicious counter threshold X. We defined K as the number of monitor nodes and P_{C} as the probability of a collision occurring in a transmission link.

When a number of alerts cross the threshold X, the rule is created and propagated to every sensor nodes by CHs. Therefore, P_{D} is the probability of more than X nodes in the total of K nodes which send an alert to CH. The event of the probability P_{D} occurs whenever there is an event that has the probability of more than X nodes sending an alert. Because the events are independent, we define P_{D} as

$$P_{\mathrm{D}} = P_{\mathrm{X}} + P_{\mathrm{X}+1} + \cdots + P_K \qquad (5)$$

DOI: 10.1002/wcm

The probability of an event that there are X nodes sending alert to CH is

$$P_{\mathrm{X}} = (1 - P_{\mathrm{C}})^{\mathrm{X}} {P_{\mathrm{C}}}^{K-\mathrm{X}} \tag{6}$$

So the probability detection of an attacker P_{D} can be written as following:

$$P_{\mathrm{D}} = (1 - P_{\mathrm{C}})^{\mathrm{X}} {P_{\mathrm{C}}}^{K-\mathrm{X}} + \cdots + (1 - P_{\mathrm{C}})^{K} {P_{\mathrm{C}}}^{K-K} \tag{7}$$

As the result, when K monitor nodes collaborate in monitoring, the probability detection of an attack is

$$P_{\mathrm{D}} = \sum_{i=\mathrm{X}}^{K} \binom{K}{\mathrm{X}} (1 - P_{\mathrm{C}})^{\mathrm{X}} {P_{\mathrm{C}}}^{K-\mathrm{X}} \tag{8}$$

We defined P_{F} as the probability of a false positive for a legitimate node. A false positive occurs in a link when a monitor node M receives a packet from D, but in its buffer does not have any information about the packet from S because of the collision. So the monitor node M may think the node D fabricating the packet instead of forwarding along the path to the destination. The monitor node considers it as a malicious action of the node D. The Figure 6 illustrates the false positive of a monitor node. The probability of false detection of monitor node M can be found as given in following steps:

$P_F = P_S + P_{\mathrm{D}}$, where P_{S} is the probability of a monitor node M which does not receive a packet from S but receive the forwarded packet from D and P_{D} is the probability of the monitor node M which receive a packet from S but does not receive the forwarded packet from D.The probability of P_{S} can be written as follows:

$$P_{\mathrm{S}} = P_{\mathrm{C}}^2 (1 - P_{\mathrm{C}}) \tag{9}$$

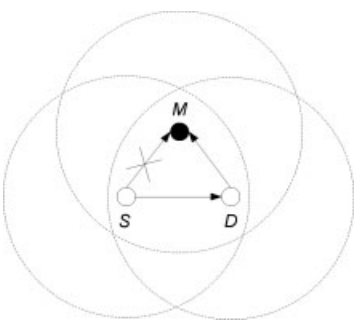

Fig. 6. False positive detection.

The probability of P_{D} can be written as follows:

$$P_D = P_C (1 - P_C)^2 \tag{10}$$

$$\Rightarrow P_{\mathrm{F}} = (1 - P_{\mathrm{C}})^2 P_{\mathrm{C}} + P_{\mathrm{C}}^2 (1 - P_{\mathrm{C}}) \tag{11}$$

Similar to Equation (8), we have the false probability of monitor nodes as

$$\Leftrightarrow P_{\mathrm{FD}} = \sum_{i=\mathrm{X}}^{K} \binom{K}{\mathrm{X}} (1 - P_{\mathrm{F}})^{\mathrm{X}} {P_{\mathrm{F}}}^{K-\mathrm{X}} \tag{12}$$

With different detection algorithms (in both wired and wireless IDS) there is always a different way to estimate the threshold. There is no way to determine the exactly threshold, just estimate and chose the best threshold based on analytical calculation of the detection algorithms and throughout simulations for the best result. In our model, the threshold is depending on the probability of collision and the average number of monitor nodes in individual transmission link, which we estimate as follow. For any two communication nodes, the average number of monitor nodes for their transmission link is the average number of sensor nodes which reside in their radio range (Figure 7).

For any distance x, the radio coverage of two communication nodes is the area of the rhombus AXBY subtracted from the area of the sectors XAY and XBY and is calculated as follows:

$$\mathrm{XY}(x) = 2r^2 \cos^{-1}\left(\frac{x}{2r}\right) - x\sqrt{r^2 - \frac{x^2}{4}} \tag{13}$$

The probability distribution function of x is given by

$$F(x) = P(\text{distance} < x) = \frac{x^2}{r^2} \tag{14}$$

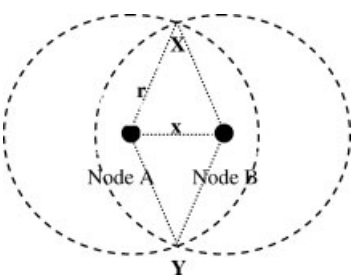

Fig. 7. The radio coverage of two communication nodes.

DOI: 10.1002/wcm

So the probability density function is

$$f(x) = F'(x) = \frac{2x}{r^2} \tag{15}$$

The expected area XY is calculated as following:

$$E[\mathrm{XY}] = \int_0^r \mathrm{XY}(x) f(x)\,\mathrm{d}x \tag{16}$$

$$\Leftrightarrow \int_0^r \left(2r^2 \cos^{-1}\left(\frac{x}{2r}\right) - x\sqrt{r^2 - \frac{x^2}{4}}\right)\frac{2x}{r^2}\,\mathrm{d}x \tag{17}$$

$$\Leftrightarrow \left(\pi - \frac{3\sqrt{3}}{4}\right) r^2 = 0.5865 r^2 \tag{18}$$

So the average number of monitor nodes for each individual link is given by $E[\mathrm{XY}] \times d$, where d is network density.

As shown in Figure 8, the scheme is effective when the number of monitor nodes is increased. The probability of a missed detection also affects the efficiency of the scheme. However, the probability of detection is close to 1, if the number of monitor nodes exceeds 5, regardless of the high probability of a missed detection. The probability of a false positive, as shown in Figure 9, indicates that the number of nodes is related to the probability of false detection. Increasing the number of nodes results in an increase in the probability of a collision. We must consider a balance between the number of monitor nodes and the probability of false detection, which suits the requirement of our applications.

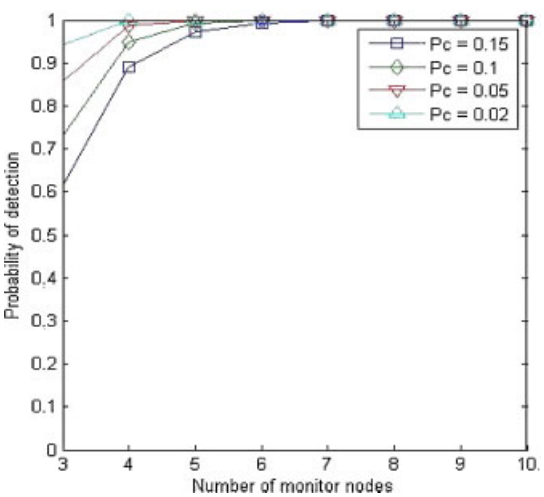

Fig. 8. Detection probability of a malicious node.

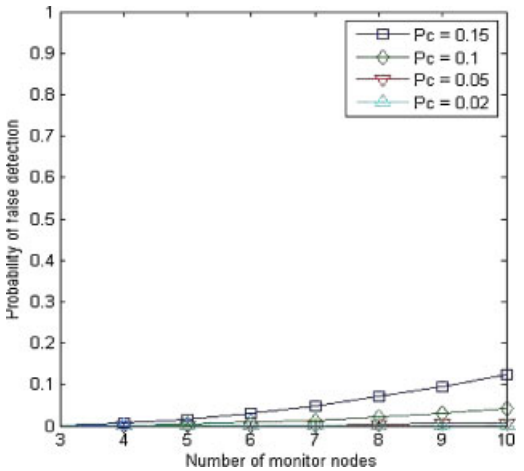

Fig. 9. False detection probability of a malicious node.

To evaluate the performance of our proposed detection scheme in realistic sensor applications, we simulate the network with 200 sensor nodes, in a field of 100 m × 100 m, using Castalia, a WSNs simulator based on Omnet++ [36]. The parameters used are in accordance with actual sensor network applications and experiments, such as Smart Dust Project (2001) [37], Virtual Patrol (2005) [38].

Sensor nodes are deployed in a randomized grid. The simple MAC carrier sense is used as the MAC protocol and simple tree routing is used as the routing protocol. The detection algorithms are implemented in the application layer. While handling the packets, the sensor nodes must call the detection algorithm before forwarding or receiving the data. To simplify the algorithms, we assign each sensor node a random trust value. There is no low-trust value during the periods of deployment.

Figure 10 shows the performance of our scheme with malicious nodes. Castalia also supports packet collision by setting the parameter SN.WirelessChannel.CollisionModel [36]. We set sensor nodes to exhibit malicious behavior by increasing their dropped packet ratio, changing the fields of forwarded packets and sending false hello packets with abnormal radio power. This result proves that our scheme yields a good packet delivery ratio under different types of routing attacks. Our simulation investigates the effect of the percentage of malicious

DOI: 10.1002/wcm

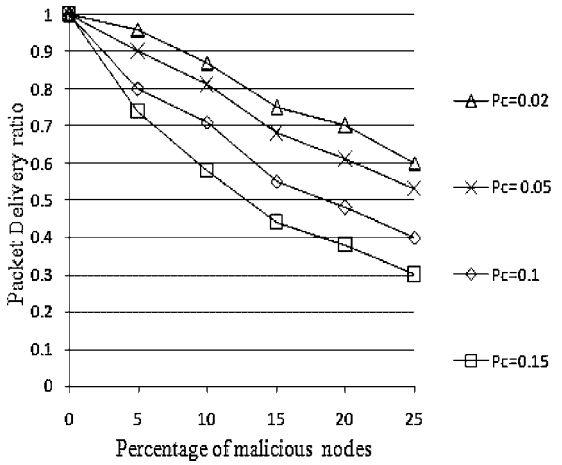

Fig. 10. Packet delivery ratio under attacks.

nodes on the packet delivery ratio. As the percentage of malicious nodes increases, revoking malicious nodes requires a particular period of time. So, the packet delivery ratio is quickly reduced, if malicious nodes increase.

As shown in Figure 11, our scheme yields a good detection rate; exceeding 90%; when the collision error is low, 2–5%, and the percentage of malicious nodes is under 5%. An increased collision ratio and malicious nodes cause greater packets loss, so it is difficult to distinguish malicious nodes and lost packets from normal nodes, because of collisions. As the collision error rate increases, misdetection is inevitable. To overcome this problem, we propose a dynamic threshold mechanism to make our scheme more efficient under a high collision rate or dropped packet rate.

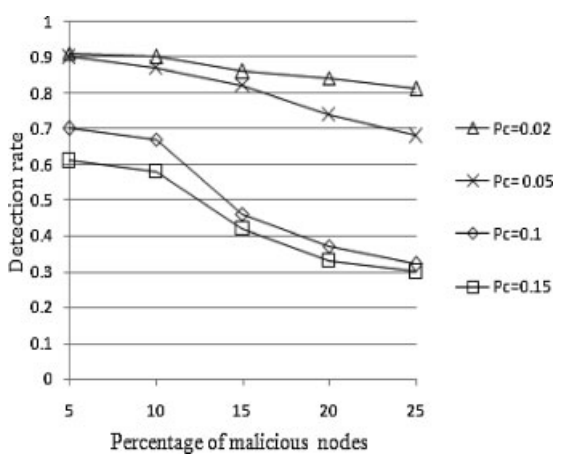

Fig. 11. Detection ratio of malicious nodes.

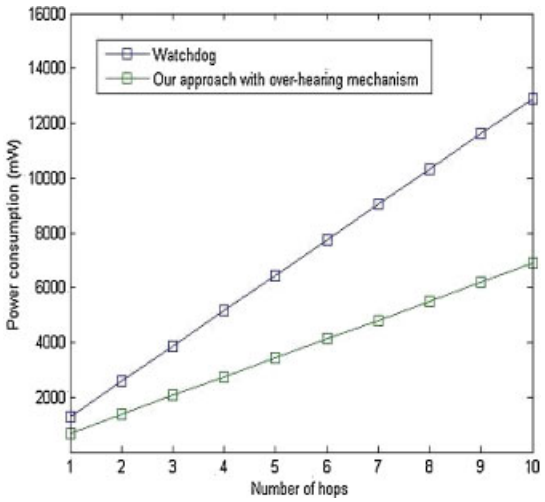

Fig. 12. Power consumption comparison.

Here, we study the energy consumption in detection modules in sensor nodes, in accordance with watchdog-based methods, and our approach with an over-hearing mechanism. Watchdog is used as a selection method of monitor nodes, which was applied in previous detection mechanisms in [8,9,20,21,25,30,33,39]. For simplicity, we analyze the energy consumption in monitor nodes in transmission from node A to node B, with n intermediate hops. Using energy consumption models in References [10,40,41], we obtain the energy consumption of monitor nodes in the transmission link in Figure 12 with various hops. It is apparent that our scheme has lower energy consumption than the watchdog-based mechanism. We postulate that our scheme reduces energy consumption in monitor nodes, thus enhances the network lifetime. In summary, in Table I we review the proposed detection framework compared with other related work on intrusion detection schemes for WSNs.

Onat and Chong's schemes are based on the model of traffic and signal power data for each neighbor node to detect anomalies. In this mechanism, as the number of neighbor nodes and sample data increase, there is substantial consumption of memory and computational resources, which results in delays in detecting attacks. Their schemes are based on previous IDS that are effective for wired networks; but we postulate it is not currently practical for WSNs. In Agah's work [26], a detection framework was proposed, based on non-cooperative games, but the detection algorithms were not shown in detail.

DOI: 10.1002/wcm

Table I. A review of related works on intrusion detection.

IDS framework	Our proposed scheme	Onat's scheme	Chong's scheme	Afrand's scheme
Characteristic				
Architecture	Distributed and collaboration	Distributed	Distributed	Distributed
Approach	Major voting, two-hop neighbor knowledge, routing rules	Traffic model and centralized detection	Traffic model and centralized detection	Non-cooperative game
Malicious nodes	High (25%)	No detail	No detail	No detail
Accuracy	High	No detail	High	Medium
Attacks	Wormhole, sinkhole, selective forwarding, and hello floods	Sinkhole	Sink hole	Denial of Service
Energy efficient	Yes	No	No	No
Delay	Medium	High	High	Medium
Memory consumption	Medium	High	High	Medium
Complex	Medium	High	High	Medium

6. Conclusion

In this paper, we propose a simple, lightweight detection framework for the prevention and detection of common routing attacks in WSNs. Our detection framework was evaluated and it was demonstrated that it was effective, even when the density of the network is high and there is a high probability of collisions in WSNs. In addition, our detection modules involve less energy consumption than techniques proposed in previous works, using an over-hearing mechanism to reduce the transmission of alert packets. In our future work, further research on this topic will be performed, with detailed simulation of different attack scenarios, to test the performance of our proposed algorithm. We expect the result to be available in the near future.

Acknowledgements

This work was supported by TTA (22008-P1-2808J45, The Development of Standard for Internet Infrastructure Security Technologies) and supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement) (IITA-2009-(C1090-0902-0002)) and by the Brain Korea 21 Program.

References

1. IlyasM, Mahgoub I. *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. CRC Press: USA, 2005.
2. Wood AD, Stankovic JA. Denial of service in sensor networks. *Computer* 2002; **35**(10): 54–62.
3. Karlof C, Wagner D. Secure routing in wireless sensor networks: attacks and countermeasures, *Ad Hoc Networks Journal, Special Issue on Sensor Network Applications and Protocols* 2003; 113–127.
4. Djenouri D, Khelladi L, Badache AN. A survey of security issues in mobile ad hoc and sensor networks. *IEEE Communications Surveys and Tutorials* 2005; **7**(4): 2–28.
5. Kown JO, Jeong IR, Lee DH. Practical password-authenticated three-party key exchange. *KSII Transactions on Internet and Information Systems* 2008; **2**(6): 312–332.
6. Mohaisen A, Nyang D, AbuHmed T. Two-level key pool design-based random key pre-distribution in wireless sensor networks. *KSII Transactions on Internet and Information Systems* 2008; **2**(5): 222–238.
7. Camtepe SA, Yener B. Key distribution mechanisms for wireless sensor networks: a survey, *Technical Report 05–07*, Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY, USA, 2005.
8. Khalil I, Bagchi S, Shroff NB. LITEWORP: a lightweight countermeasure for the wormhole attack in multihop wireless networks, *International Conference on Dependable Systems and Networks*, 2005.
9. Khalil I, Bagchi S, Shroff NB. MOBIWORP: mitigation of the wormhole attack n mobile multihop wireless networks. *Proceeding of Securecomm and Workshops*, *2006*.
10. Sultana N, Huh E-N. Application driven cluster based group key management with identifier in wireless sensor networks. *KSII Transaction on Internet and Information System* 2007; **1**(1): 5–18.
11. Heady R, Lugar G, Servilla M, Maccabe A. The architecture of a network level intrusion detection system, *Technical Report*, Computer Science Department, University of New Mexico, 1990.
12. Roesch M. 2002. The SNORT network intrusion detection system, http://www.snort.org
13. Paxson V. BRO: a system for detecting network intruders in real-time. *Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX, USA, 1998.
14. Balasubramaniyan JS, Garcia-Fernandez JO, Isacoff D, Spafford EH, Zamboni D. An architecture for intrusion detection using autonomous agents. *The 14th Annual Computer Security Applications Conference (ACSAC' 98)*, Scottsdale, AZ, USA, 1998.

DOI: 10.1002/wcm

15. Cuppens F, Miege A. Alert correlation in a cooperative intrusion detection framework. *Proceedings of IEEE Symposium on Security and Privacy*, 2002.
16. Janakiraman R, Waldvogel M, Zhang Q. Indra: a peer-to-peer approach to network intrusion detection and prevention. *Proceedings of IEEE WETICE*, 2003.
17. Aboelaze M, Aloul F. Current and future trends in sensor networks: a survey, *Second IFIP International Conference on Wireless and Optical Communications Networks*, 2005.
18. Crossbow Technology, Inc. MICA2, wireless measurement system, http://www.xbow.com
19. Mishra A, Nadkarni A, Patcha K. Intrusion detection in wireless ad hoc networks. *IEEE Wireless Communications* 2004; **11**(1): 48–60.
20. Roman R, Zhou J, Lopez J. Applying intrusion detection systems to wireless sensor networks. *The 3rd IEEE CCNC*, 2006.
21. Marti S, Giuli TJ, Lai K, Baker M. Mitigating routing misbehavior in mobile ad hoc networks. *Proceedings of MOBICOM 2000*, pp. 255–265, 2000.
22. Onat I, Miri A. An intrusion detection system for wireless sensor networks. *IEEE International Conference on Wireless And Mobile Computing, Networking And Communications*, 2005.
23. Banerjee S, Grosan C, Abraham A. IDEAS: intrusion detection based on emotional ants for sensors. *The 5th International Conference on Intelligent Systems Design and Applications*, 2005.
24. Techateerawat P, Jennings A. Energy efficiency of intrusion detection systems in wireless sensor networks. *IEEE/WIC/ACM International Conference on Web Intelligence and International Agent Technology Workshops*, 2006.
25. Loo CE, Ng MY, Leckie C, Palaniswami M. Intrusion detection for routing attacks in sensor networks. *International Journal of Distributed Sensor Networks* 2006; **2**(4): 313–332.
26. Silva A, Loureiro AAF, Martins M, Ruiz LB, Rocha B, Wong H. Decentralized intrusion detection in wireless sensor networks. *International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, October 2005.
27. Agah A, Basu K, Das SK. Security enforcement in wireless sensor networks: a framework based on non-cooperative games, *Pervasive and Mobile Computing* 2006; **2**(2): 137–158.
28. Wang Y, Attebury G, Ramamurthy B. A survey of security issues in wireless sensor networks. *IEEE Communication Surveys* 2006; **8**(2): 2–23.
29. Abbasi AA, Younis M. A survey on clustering algorithms for wireless sensor networks. *Computer Communications* 2007; **30**(14-15): 2826–2841.
30. Hu J. Cooperation in mobile ad hoc networks, *Technical Report*, 2005.
31. Durresi A, Paruchuri V, Iyengar SS, Kannan R. Optimized broadcast protocol for sensor networks. *IEEE Transaction on Computers* 2005; **54**(8): 1013–1024.
32. Debar H, Curry D, Feinstein B. The intrusion detection message exchange format. Internet experimental RFC 4765, March 2007. http://tools.ietf.org/html/rfc4765
33. Kaplantzis S, Shilton A, Mani N, Sekercioglu YA. Detecting selective forwarding attacks in wireless sensor networks using support vector machines. *The 3rd IEEE International Conference on ISSNIP*, 2007.
34. Bianchi G. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications* 2000; **18**(3): 535–547.
35. Le H.C, Herve G, Noureddine Z. Overhearing for energy efficient in event-driven wireless sensor networks. *Proceeding of First IEEE International Workshop on Intelligent System Techniques*, 2006.
36. Castalia simulator, http://castalia.npc.nicta.com.au
37. Pister K, Boser B, Kahn J. Smart dust project. 2001. http://robotics.eecs.berkeley.edu/~pister/SmartDust/.
38. Gui C, Mohapatra P. Virtual patrol: a new power conservation design for surveillance using sensor networks. *Fourth International Symposium on Information Processing in Sensor Networks*, 2005.
39. Hai TH, Huh E-N. Hybrid intrusion detection for wireless sensor networks. *Proceeding of the ICCSA*, 2007.
40. Hai TH, Huh E-N. Optimal selection and activation of intrusion detection agents for wireless sensor networks. *Proceeding of International Conference on Future Generation Communication and Networking*, 2007.
41. Holger K, Adreas W. (eds). Energy consumption on sensor nodes. In *Protocols and Architecture for Wireless Sensor Networks*. John Wiley & Sons Press: Germany, 2005.
42. Hill J, Szewczyk R, Woo A, *et al*. System architecture directions for networked sensors. *Proceeding of 9th Inter national Conference on ASPLOS* 2000; **28**(5): 93–104.
43. Hill J, Szewczyk R, Woo A, *et al*. System architecture direction for networked sensors. *ACM SIGOPS Operating Systems Review* 2000; **34**(5): 93–104.

Authors' Biographies

Tran Hoang Hai has earned B.S. degree from Hanoi University of Technology in Vietnam and Master's degree in Computer Engineering from Kyung Hee University, South Korea in 2008. He is currently a Ph.D. student in DIONYSOS project, INRIA, France since December 2008. His interesting research areas are wireless security, sensor network, and applied game theory to communication network.

Eui-Nam Huh has earned B.S. degree from Busan National University in Korea, Master's degree in Computer Science from University of Texas, USA in 1995 and Ph.D. degree from the Ohio University, USA in 2002. He was a Chair of Grid Project Standard Group for 2006–2007, TTA, Korea. He is steering committee and workshop chair of ICCSA. He is one of the editors of the *Journal of Korean Society for Internet Information* and *KSII Transactions on Internet Information Systems*. He was also an Assistant Professor in Seoul Women's University, South Korea. Now he is working as a Professor in the Department of Computer Engineering, Kyung Hee University, South Korea. His interesting research areas are high performance network, cloud computing, sensor network, security, distributed real time system, middleware, and sports engineering.

DOI: 10.1002/wcm

Minho Jo received the B.S. degree in industrial engineering from Chosun University, Seoul, South Korea, and the Ph.D. degree in computer networks from the Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA in 1994. He worked as a Staff Researcher with Samsung Electronics, South Korea and was a Professor at the School of Ubiquitous Computing and Systems, Sejong Cyber University, Seoul. He is now a Research Professor at the Graduate School of Information Management and Security, Korea University, Seoul, South Korea. Professor Jo is serving as Executive Director of the Korean Society for Internet Information (KSII) and Board of Trustees of the Institute of Electronics Engineers of Korea (IEEK), respectively. He is Founding Editor-in-Chief and Chair of the Steering Committee of *KSII Transactions on Internet and Information Systems*, General Chair of International Ubiquitous Conference, and Co-Chair of the International Conference on Ubiquitous Convergence Technology. He is one of the editors of the *Journal of Wireless Communications and Mobile Computing* and Associate Editor of the *Journal of Security and Communication Networks* published by Wiley, respectively. He serves as an Associate Editor of the *Journal of Computer Systems, Networks, and Communications* published by Hindawi. He served as Chairman of IEEE/ACM WiMax/WiBro Services and QoS Management Symposium, IWCMC 2008. He is Technical Program Committee of IEEE ICC 2008 & 2009 and IEEE GLOBECOM 2008 & 2009 and TPC Chair of CHINACOM 2009 Network and Information Security Symposium. His current interests lie in the area of wireless sensor networks, RFID, wireless mesh networks, security in communication networks, machine intelligence in communications, and ubiquitous and mobile computing.

DOI: 10.1002/wcm