# An Experimental Study of Hierarchical Intrusion Detection for Wireless Industrial Sensor Networks

Sooyeon Shin, Taekyoung Kwon, *Member, IEEE*, Gil-Yong Jo, Youngman Park, and Haekyu Rhy, *Member, IEEE*

***Abstract*—Wireless industrial sensor networks are necessary for industrial applications, so that wireless sensor nodes sense around themselves and detect anomaly events in the harsh industrial environments. Due to the harshness, anomaly events such as adversarial intrusions may result in harmful and disastrous situations for industrial applications but it is difficult to detect them over wireless medium. Intrusion detection is an essential requirement for security, but as far as we know, there have not been such studies for wireless industrial sensor networks in the literature. The previous intrusion detection methods proposed for wireless sensor networks consider networks rather in general senses and restrict capabilities to specific attacks only. In this paper, we first study intrusion detection for wireless industrial sensor networks, through various experiments and design of a hierarchical framework. We classify and select better methodologies against various intrusions. Subsequently, we find novel results on the previous methodologies. We also propose a new hierarchical framework for intrusion detection as well as data processing. Throughout the experiments on the proposed framework, we stress the significance of one-hop clustering, which was neglected in the previous studies. Finally, we construct required logical protocols in the hierarchical framework; hierarchical intrusion detection and prevention protocols.**

***Index Terms*—Clustering, industrial applications, intrusion detection, intrusion prevention, wireless industrial sensor network.**

## I. Introduction

### A. Background

IN VARIOUS industrial environments, sensors and their networks are deployed for sophisticated sensing and control purposes. In the harsh environments, however, we usually experience a great number of hazards that can range from strong mechanical vibrations, high temperatures, fragile surfaces, noisy electrical affects, and even explosive gases. Though wired industrial communications, such as Fieldbus systems and wired HART have been installed successfully in the field of Factory Automation and Process Automation [33], it is still difficult and expensive to install wiring due to the harshness and complexity of those environments. Thus, there are strong needs for wireless communication technologies to be applied for sensing and alerting those environments [9], [35].

WISNs, which stands for Wireless Industrial Sensor Networks, have emerged as perfect technological solutions to those needs [3], [14], [18], [19], [22], [27], [29]. The WISN is one of the most sensitive types of Wireless Sensor Networks (WSNs). WISNs are distributed wireless networks of tiny sensor nodes which may sense around themselves and report their results, such as emergency alerts, in a timely and reliable manner through wireless multihop connections in the industrial environments.

### B. Motivation

There are several difficulties in managing those wireless networks including WSNs effectively and detecting various types of adversarial intrusions over wireless medium in the harsh environments. First of all, the use of wireless communications medium may allow various attacks such as eavesdropping, illegal modification, and fabrication more easily than the wired medium [9], [14]. It is not easy to detect various types of adversarial intrusions over wireless medium without monitoring all communication traffics as well as sensor nodes, which is quite impractical. Even worse, those adversarial intrusions may result in harmful and disastrous events specifically for industrial environments. If the critical status of proprietary mechanics is eavesdropped and monitored by competitors through WISNs, there can be enormous losses in the economical senses. If there is no timely and reliable alert from WISNs saying if an alert is stolen or modified, the leakage of toxic chemicals, radiations, flammable liquids, and gases could pollute the environment and endanger the public [22]. Therefore, the vulnerability of WISNs to various kinds of attacks should be of great concern.

WISNs should be robust and self-repairing against adversarial intrusions thus guaranteeing fail-safe operations of industrial applications including their equipments. For this purpose, cryptographic methods for WSNs, such as encryption, authentication and key management, can be employed. However, most methods have focused on prevention techniques as the first line of defense. There is a non-negligible probability that an active attacker will succeed in launching some attacks that of which have no known prevention methods or have not been experienced before. As a result, WISNs cannot depend on just prevention techniques alone. It is necessary to use an intrusion detection mechanism as the second line of defense in case prevention fails. Due to the extreme sensitiveness of industrial environments including target devices, it is required to detect possible intrusions very effectively, meaning fast and accurate. Since malicious events through various types of adversarial intrusions may compromise WISNs, as well

Manuscript received July 09, 2009; revised October 21, 2009, January 28, 2010, March 20, 2010; accepted April 29, 2010. Date of publication September 02, 2010; date of current version November 05, 2010. This work was supported in part by a grant from the National Research Foundation of Korea funded by the Korean Government (2009-0077066) and in part by KT Future Technology Laboratory and in part by The Ministry of Knowledge Economy (MKE), Korea, under the ITRCsupport program supervised by the National IT Industry Promotion Agency (NIPA) (NIPA-2010-C1090-1001-0004). Paper no. TII-09-07-0143.

S. Shin, T. Kwon, and G. Jo are with the Department of Computer Engineering, Sejong University, Seoul 143-747, Korea (e-mail: shinsy80@sju.ac.kr; tkwon@sejong.ac.kr; ggkill@sju.ac.kr).

Y. Park and H. Rhy are with KT Future Technology Laboratory, Seoul 137-792, Korea (e-mail: youngman@kt.com; rhg@kt.com).

Digital Object Identifier 10.1109/TII.2010.2051556

as other coexistent systems and networks, and result in more severe disasters around industrial environments, it is inevitable that WISNs require intrusion detection mechanisms than usual WSN configurations.

To the best of our knowledge, Intrusion Detection Systems (IDSs) have been studied sporadically in the literature of WSNs by aiming at different attacks and features respectively, e.g., [2], [15], [21], [23], [26], [28], and [30], but not for WISNs. Of course, most techniques including intrusion detection mechanisms for WSNs can be applied to WISNs [14]; however, the previous IDSs studied for WSNs considered specific kinds of attacks for their detection capabilities or designed general frameworks only. Thus, they are not very suitable for WISNs in generic perspectives. It is required to compare those methods with respect to detection capabilities and to provide more specific framework for accommodating them in WISNs, of which many industrial applications including equipment monitoring, environment monitoring, and industrial automation may be susceptible to different kinds of attacks. We need more tight construction of IDSs over the WISN than in the previous studies.

### C. Contribution

In this paper, we first study intrusion detection for WISNs in the way of achieving the goals described above. First of all, we analyze and compare previous IDSs of WSNs empirically through various experiments using real sensor nodes. Based on the result of experiments, we classify and select better methodologies against various attacks that must be harmful to industrial environments. For example, we conduct experiments on three kinds of detection techniques against a selective forwarding attack; one using the so-called "interval rule" [28], another using the predefined specification of this attack [15], and lastly by checking packet dropping [31]. Through these experiments, we conclude that the last one has the highest detection rate, so that it can be used as the best detection technique against the selective forwarding attack. Subsequently with these experiments, we disclose several unknown features of the previous detection techniques. For instance, on the previous techniques targeting a packet jamming attack, we find out that this attack can actually drop packets as an Radio Frequency (RF) jamming attack [37] does rather than flooding an enormous number of packets onto nodes nearby. It implies that a detection technique against this attack has to detect an abrupt decrease of receiving rate as well as an increase of it sensitively, while the previous techniques only consider the receiving rate's increase beyond of the average packet arrival rate or predetermined threshold.

Second, we propose a new hierarchical framework for intrusion detection as well as data processing in WISNs. In general, sensor nodes can detect intrusions using the broadcast nature of transmission within one-hop. It is unfeasible for sensor nodes to monitor malicious neighbors outside one-hop perfectly, even if they have help from intermediates. Thus, one-hop clustering is necessary for efficient intrusion detection in WISNs. Throughout the experiments on the proposed framework, we derive a result on the significance of one-hop clustering, which must be necessary for wireless industrial networks while the existing hierarchical IDSs rely on the conventional clustering methods [1], [12], [13], [32], [38] without guaranteeing one-hop clustering. On the other hand, multihop clustering is also necessary for effective data processing in WISNs. Thus, appropriate care must be taken to forming node clusters in WISNs, in the way of considering both intrusion detection and data aggregation. For this purpose, we construct a hierarchical framework based on two-level clustering; multihop clusters for data aggregation (the first clustering) and one-hop clusters for intrusion detection (the second clustering). Through our hierarchical framework, we allow WISNs to perform in-network processing, so that industrial applications can obtain more accurate results regarding of sensing and intrusion monitoring and save energy in WISNs. Finally, we construct required logical protocols in the proposed hierarchical framework; a hierarchical intrusion detection protocol and intrusion prevention protocol.

Our framework including intrusion detection and prevention protocols satisfies flexibility and reliable transmission as requirements for WISNs. For flexibility, WISNs can configure itself based on two-level clustering and repair itself using IDS modules of sensor nodes. The hierarchical intrusion detection protocol allows employing the classified intrusion detection techniques to differentiate selecting according to the industrial applications on the basis of various experiments. This protocol also allows to add new detection techniques easily to an IDS module for the future without modifying the protocol. For reliable transmission, our intrusion prevention protocol utilizes different key establishment with regard to the cases of deployment of WISNs and establishes different types of keys according to the role of a sensor node. The prevention protocol also enables to encrypt a message selectively or to append a message authentication code to its related critical proprietary information of industrial applications. Our protocols also satisfy real-time communications, a typical requirement for the industrial applications. Symmetric cryptography operations and the detection techniques, which can be used for our prevention and detection protocols respectively, may have negligible influence on the real-time performance of industrial applications. Especially, for showing the influence of the detection techniques, we additionally perform an experiment to evaluate the amount of time required for their executions. Note that if any intrusion is detected, the management of discovered intrusion must have the highest priority. The influence on the performance resulting from this task is less important.

### D. Organization

The rest of this paper is organized as follows. Section II describes preliminaries. Section III introduces a new hierarchical framework based on two-level clustering, with intrusion detection and prevention protocols. Section IV shows the results of experiments on the existing IDSs, as well as the necessity of one-hop clustering for intrusion detection. Finally, Section V presents the conclusion of this work.

## II. Preliminaries

In this section, we briefly describe the related work on industrial applications, the previous intrusion prevention and detection methods. We also point out problems of the existing hierarchical IDSs with respect to WISNs.

### A. Related Work

*1) Industrial Applications:* WISNs are necessary for many industrial applications, which include industrial monitoring and industrial automation and management. In industrial monitoring applications, WISNs can be used to monitor the maintenance of equipments and machineries remotely. Predictive maintenance [19] is the general term that allows the user to detect machine failures and to reduce repair cost. Intel's EcoSense project group is employing a preventive maintenance application in which a WISN is used for monitoring the health of semiconductor fabrication equipment [22]. WISNs can be also useful for environmental monitoring such as climate reporting, leakage detection and so on. Especially, sensor nodes play an important role to monitor leakage of chemicals and radiation since leakage of toxic chemicals, radiation, flammable and explosive liquids and gases can have a hazardous influence upon the environments nearby. For preventing this kind of danger, many oil and gas companies are considering to deploy WISNs widely [22]. WISNs are essential to industrial automation and management. In this general class, sensor nodes link control systems with the physical processes. In inventory management systems [22], WISNs improve the visibility of materials and enable the user to manage and control real-time inventory data. For example, BP uses a WISN to monitor its industrial customers' LPG tank remotely [22].

As mentioned previously, in WISNs, it is necessary to consider typical requirements, such as timely, reliable communication. Lost or delayed data may cause the above industrial applications to malfunction. The vulnerabilities of WISNs to various kinds of attacks should be also great concern. In our view, technical details of such attacks can be generalized with regard to WSNs, for example, eavesdropping, impersonation, and infiltration over wireless channels, as well as physical node capture. However, their consequences must be more tragic in the WISN than in the usual WSNs in the both economical and environmental aspects. An adversarial principal may capture a certain sensor node, and then extract a common secret key to eavesdrop valuable data being sensed and exchanged over the industrial environment, e.g., industrial espionage. The adversary can transmit bogus commands to neighboring sensor nodes or fabricated data to base stations over wireless channels. This attack causes the corresponding industrial devices to malfunction, e.g., overheat themselves. Thus, WISNs should be robust and self-repairing against adversarial intrusions, and guarantee fail-safe operation of their applications and equipments.

*2) Intrusion Prevention Mechanisms:* In WISNs, sensor nodes are susceptible to various security attacks due to the broadcast nature of the transmission medium, the limited resources and their dangerous or unapproachable deployment. First of all, to protect sensor nodes against such attacks, TinySec and ZigBee security protocols are used as basic security mechanisms. TinySec [17] provided by TinyOS [44] is a lightweight link layer security mechanism, which supports symmetric key encryption using cipher block chaining (CBC) and authentication using a message authentication code (MAC). Sensor nodes such as Tmote sky [42] or Crossbow motes [41] can use IEEE 802.15.4/ZigBee [45] as the wireless technology [5]. The IEEE 802.15.4/ZigBee is designed for a low-cost, standard-based and flexible wireless network technology, which offers low power consumption, reliability, interoperability and security for control and monitoring applications. ZigBee supports Advanced Encryption Standard (AES) encryption with a 128 bits key and data integrity using a MAC. However, basic security mechanisms have several vulnerabilities. TinySec is susceptible to physical attacks [20]. In ZigBee residential mode, all sensor nodes share one secret key and the whole network can be compromised if an attacker achieves it. In ZigBee commercial mode, the network is susceptible to single point of failure because one Trust Center manages every key of the network.

Second, numerous cryptographic mechanisms such as key management, secure routing, and so on, have been proposed to ensure the security of network services and applications in WSNs. They can be also used to protect WISNs. Especially, key management is a core mechanism for ensuring data confidentiality, data and node authentication, data integrity, etc. While recent studies have shown that public key cryptography such as RSA and Elliptic Curve Cryptography (ECC) is feasible in WSNs, it is still expensive for the most applications in WSNs and WISNs [34]. Thus, most studies [10], [6], [8], [39] have focused on symmetric key cryptography in WSNs. Key management protocols can be divided into probabilistic and deterministic key protocols according to the probability of key sharing. Most of them use probabilistic approaches that select randomly several keys from a large key pool and load them to each node. These approaches have disadvantages that some nodes have no shared keys with their neighbors at all and they cannot establish secure link with them. On the contrary, in deterministic key protocols, all nodes can establish keys with neighbors using few preloaded keys. With respect to the deployment of WISNs in industrial environments, it is often impossible to preload keys due to possible obstructions and deployment errors. In this case, sensor nodes should dynamically establish keys with their neighbors after deployment using key establishment protocols for WSNs. Thus, deterministic key protocols are more suitable for WISNs due to the harshness of industrial environments and their efficiency regarding of storage and computing costs. In Localized Encryption and Authentication Protocol (LEAP) [39] based on the deterministic approach, a node can establish pairwise keys with any immediate neighbors using an initial key preloaded before deployment. Therefore, the single initial key is only necessary for pairwise key establishment in LEAP, compared to a number of preinstalled keys in probabilistic key protocols.

As we mentioned above, various security and cryptographic mechanisms of WSNs can be used for securing WISNs against various attacks. However, most of them have focused on prevention techniques as the first line of defense. Of course, they can protect WISNs from various types of passive and outside attacks, but there is a non-negligible probability that an attacker will success to launch active and inside attacks. Even worse, there can be remained some attacks that have no known prevention methods and have not been experienced before. For example, an active attacker can launch a physical attack that has physical access to a sensor node and extracts sensitive information such as cryptographic keys. Subsequently, (s)he can alter

or replicate the compromised node and reinject it or its clones into the network for further attacks. In this case, prevention techniques such as encryption and authentication with cryptographic keys are useless. As a result, WISNs cannot depend on prevention techniques alone, and thus it is necessary to accommodate intrusion detection mechanisms as the second line of defense when prevention fails.

*3) Intrusion Detection Systems:* IDS architectures are divided into three basic categories with regard to the detection technique: misuse detection, anomaly detection [34], and specification-based IDS, introduced by Brutch and Ko [4]. IDS architectures for WSNs can further be classified into three categories like ad-hoc networks' IDS architectures [4] according to network structures: a standalone IDS [21], [28], [2], distributed/cooperative IDS [23], [15], [26] and hierarchical IDS [31], [25], [30], [26]. In the standalone IDSs, each node has equipped with an IDS agent, runs it independently without exchanging any information with other nodes and it is responsible for detecting intrusions by itself. The distributed/cooperative IDSs are similar to the standalone IDSs except that each node cooperates with its neighbors. In the hierarchical IDSs, the extended version of the distributed/cooperative IDSs, all nodes have equipped with an IDS agent and they detect intrusions locally. The hierarchical IDSs have the hierarchy that consists of cluster heads and member nodes. In other words, the network is divided into several clusters that have cluster heads. Each cluster head is responsible for monitoring its own member nodes' packets and alerting the network to intrusions.

To the best of our knowledge, there is no intrusion detection systems specific for WISNs in the literature. Due to the extreme sensitiveness of environments (including such target devices), it is required significantly to detect possible intrusions very effectively, meaning fast and accurate, in the WISN. However, the previous IDSs devised for WSNs are not very suitable for such cases and have several problems to be applied to WISNs as matters stand. First, most of them only considered the specific attacks and the corresponding detection techniques, or general architectures. Second, despite that one-hop clustering is an essential requirement for detecting possible intrusions directly in WISNs, the previous IDSs rely on the conventional clustering methods which does not guarantee one-hop clustering. Thus, industrial applications need more tight construction of IDSs that can monitor directly and handle various attacks against WISNs.

### B. Problems of Previous Approaches

We investigate two problems of the previous approaches, especially the hierarchical IDSs. First, they only considered specific attacks and the corresponding detection techniques, or general architectures (*Problem I*). Second, the existing hierarchical IDSs rely on the conventional clustering methods without guaranteeing one-hop clustering which is essential for efficient intrusion detection in WISNs (*Problem II*).

*1) Problem I:* Industrial applications are susceptible to the different type of attacks. Most of the existing IDSs, however, restricted their capabilities to specific attacks only. [23] considered intrusion detection methods for two types of attacks; node impersonation and resource depletion, and [2] considered the detection technique against the forged packet attack. On the other hand, several IDSs proposed general architectures only. [26] proposed only general guidelines of an IDS architecture. Various detection techniques should be considered for many kinds of industrial applications. It is also required to compare those methods with respect to detection capabilities and to provide more specific architecture for accommodating them.

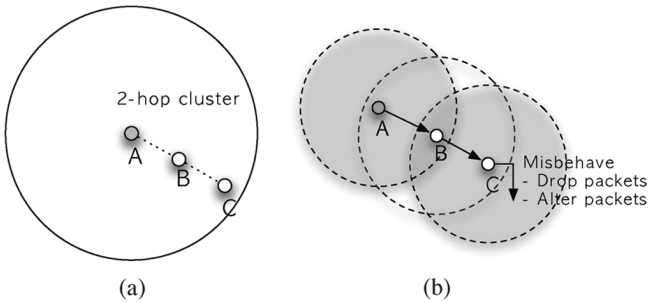


Fig. 1. Problems of previous hierarchical IDSs. (A solid circle indicates a two-hop cluster and dashed circles indicate communication ranges.) (a) Two-hop cluster. (b) Example for problems.

*2) Problem II:* Intrusion detection in most of the existing IDSs is operated within one-hop using the broadcast nature of transmission (i.e., the watchdog approach [15]). A sensor node can detect misbehavior of nodes only within one-hop. However, the well-known clustering algorithms (i.e., [13]), the previous hierarchical IDSs employ, do not guarantee one-hop clustering. Namely, it is difficult for a cluster head to monitor member nodes directly outside of its communication range. In Fig. 1(a), suppose that A is a cluster head that monitors its member nodes, while B and C are its member nodes in the two-hop cluster. In case that C misbehaves, the cluster head A cannot detect it without intermediate node B's help because C is out of A's communication range. For example, in Fig. 1(b), suppose that B and C are intermediates for transmitting a packet to the base station and C is a malicious node which tries to drop the packet. A is able to check whether B relays the packet, while it is impossible for A to monitor and check directly whether C relays the packet.

Naturally, a cluster head can use indirect monitoring in the way of solving the problems described above. In indirect monitoring, the node can monitor misbehavior nodes outside of one-hop by the help of neighbors located within one-hop. However, communication overhead will incur and the reliability of intermediates, such as the sleep rate and error rate, has influence on the intrusion detection rate.

To illustrate the effects of reliability of intermediates on intrusion detection, we give an example using the sleep rate and error rate. We consider a cluster in the clustered network. Suppose that there is one malicious node and its cluster head is responsible for detecting the malicious node. Also, suppose that all nodes including a cluster head can always monitor the malicious node's traffic within own communication range due to direct monitoring. The cluster head detects the malicious node within its communication range by itself or outside of one-hop by help of a member node who is located at one-hop from the malicious node. To help the cluster head, member nodes monitor all traffic within their own communication range, and then

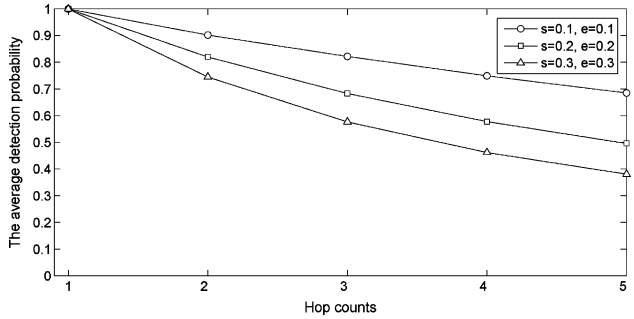


Fig. 2. Average detection probabilities according to hop counts, considering sleeping rate and error rate.

transmit the results of monitoring to the cluster head. We assume that all nodes except the cluster head and malicious node have the same sleep probability $s$ that a node is in the sleep state with $0 \leq s \leq 1$ and the same error probability $e$ that packet loss on the sending nodes occurs with $0 \leq e \leq 1$. We consider the detection probability $p_{i,j}$, with $1 \leq j \leq i$, that a cluster head detects a malicious node located at $j$ hop from the cluster head in a $i$-hop cluster.

We first consider a one-hop cluster. Suppose that a malicious node [i.e., the node B in Fig. 1(b)] is located within the cluster. In this case, a cluster head [i.e., the node A in Fig. 1(b)] can always detect it regardless of the cluster head's $s$ and $e$. Thus, $p_{1,1}$ is 1. Second, we consider a two-hop cluster. Suppose that a malicious node can be located at one-hop or two-hop from a cluster head. In the former case, $p_{2,1}$ is the same as $p_{1,1}$. On the other hand, in the latter case, for detecting it, the cluster head depends on the help of an intermediate [i.e., the node B in Fig. 1(b)] located at one-hop from the malicious node and cluster head [i.e., the node C and A in Fig. 1(b), respectively]. Thus, the intermediate's $s$ and $e$ have an effect on $p_{2,2}$ and $p_{2,2}$ is $(1-s)(1-e)$.

Let $P_i$ be the average detection probability that a cluster head detects a malicious node as an intruder at anywhere within the cluster of $i$-hop clusters. In the case of the two-hop cluster, $P_2$ is $(p_{2,1}+p_{2,2})/2$, such that $\{1+(1-s)(1-e)\}/2$. Considering a three-hop cluster and a malicious node located at one-hop, two-hop, or three-hop from the cluster head, $p_{3,1}$ and $p_{3,2}$ are the same as $p_{1,1}$ and $p_{2,2}$, respectively. In the last case, $p_{3,3}$ may be influenced by $s$ and $e$ of both two intermediates at one-hop and at two-hop, so that $p_{3,3}$ is $\{(1-s)(1-e)\}^2$. In the $i$-hop clusters, if there is a malicious node at $j$-hop where $1 \leq j \leq i$, we have $p_{i,j} = \{(1-s)(1-e)\}^{j-1}$. Especially, $p_{i,1}$ is 1 since the cluster head can always detect the malicious node within its own communication range; if $j = 1$, the computed value of the above formula matches 1. Thus, the average detection probability $P_i$ of the $i$-hop cluster is as follows:

$$P_i = \frac{1}{i}\sum_{j=1}^{i} p_{i,j}. \tag{1}$$

Fig. 2 describes the average detection probability $P_i$ decreases according to the increase of hop counts. In other words, indirect monitoring has a bad effect upon intrusion detection. In conclusion, direct monitoring within one-hop clusters should be always possible and one-hop clustering must be necessary for efficient intrusion detection in industrial applications. In Section IV-C, we show the significance of one-hop clustering through experimenting on the above example with real motes.

## III. Intrusion Detection and Prevention Based on Two-Level Clustering

In this section, we construct a hierarchical framework on the basis of two-level clustering. We construct required logical protocols in the proposed hierarchical framework; an intrusion detection protocol and intrusion prevention protocol. WISNs are often deployed in the harsh industrial environments. Since it is impractical for an administrator to intervene timely when malicious events occur through various types of adversarial intrusions, WISNs should be robust and self-repairing. For satisfying these requirements, every sensor node in our detection protocol estimates intrusions by itself using its IDS module and handles them according to its role (i.e., a gateway and cluster head). Furthermore, the typical requirements for WISNs, such as real-time, reliable communication, should be considered [14]. The detection protocol with the hierarchical framework enables WISNs to serve a timely and reliable warning on their industrial applications and systems. In the hierarchical intrusion prevention protocol, it is feasible to transmit sensing and detecting results in a timely and reliable manner through in-network processing and prevention mechanisms such as encryption and message authentication codes, respectively. Besides, both detection techniques and symmetric cryptography algorithms adopted for intrusion detection and prevention spend less time executing them. Thus, our protocols may satisfy the typical requirements.

### A. Our Resolution: Two-Level Clustering

As we mentioned, it is positively necessary for efficient intrusion detection in industrial environments to form one-hop clusters. On the other hand, one-hop clusters are not practical in terms of data gathering and energy efficiency. Thus, care must be taken to form node clusters in WISNs, regarding both intrusion detection and data gathering. For this purpose, we construct the hierarchical framework based on two-level clustering which consists of the first clustering (multihop clustering) for efficient data gathering and the second clustering (single-hop clustering) for effective intrusion detection.

Fig. 3(a) presents an example of clusters formed by two-level clustering, and Fig. 3(b) shows the hierarchies as a result of two-level clustering and logical relationship between hierarchies; the base station (BS), gateways (GWs), cluster heads (CHs), and member nodes (MNs). To distinguish cluster heads of the first clustering from cluster heads of the second clustering, we will refer to a cluster head of the first clustering as GW and that of the second clustering as CH.

In the industrial environments, we can consider at least three cases of deployment with regard to a WISN. First, when the WISN is deployed in new industrial environments with careful deployment plan and construction, sensor nodes might be positioned at fixed and well-planned locations. Second, when the WISN is deployed in the existing industrial environments with additional deployment plans, sensor nodes might be located flexibly due to possible obstructions and deployment

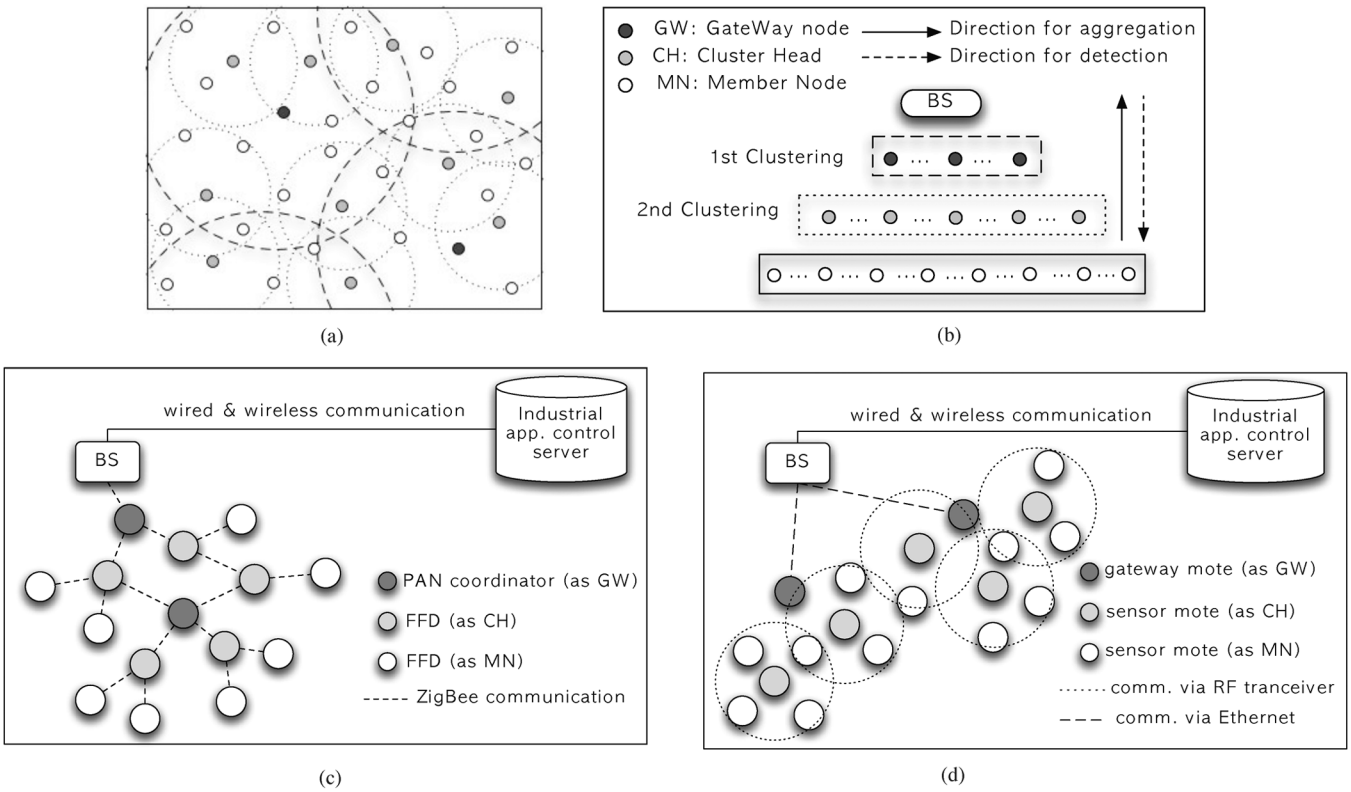


Fig. 3. A hierarchical framework based on two-level clustering. (a) Two-level clusters. (b) Hierarchies based on two-level clustering. (c) Two-level clustering based on ZigBee. (The cluster tree topology of ZigBee is adopted.) (d) Two-level clustering based on the heterogeneity of motes.

errors. Third, when the WISN is deployed in the existing harsh industrial environments, sensor nodes might be scattered in quite random manners and located dynamically with possible deployment errors. Since two-level clustering is a logical concept of hierarchy, we do not aim to propose a new two-level clustering algorithm. Two-level clustering is not limited to the specific methods, but the existing standard technologies (i.e., IEEE 802.15.4/ZigBee) and cluster algorithms [1], [7], [12], [13], [32], [38] can be utilized according to applications of WISNs.

Let us discuss more on these three cases with respect to the possible integration methods. In the first and the second cases with careful deployment plans, industrial applications can adopt the IEEE 802.15.4/ZigBee standard [45] or the heterogeneity of motes for two-level clustering. IEEE 802.15.4 distinguishes two types of nodes: Full Function Devices (FFD) and Reduced Function Devices (RFDs). FFD can play three kinds of roles: as a PAN coordinator, as a coordinator or as a device, while RFD can only play the role of device. Sensor nodes are suitable for both FFD and RFD operation. ZigBee also supports several network topologies: star, mesh and cluster tree. It is possible for industrial applications to deploy the homogeneous WISN according to the mesh and cluster tree topologies using ZigBee sensor nodes equipped with the IEEE 802.15.4 compliant RF transceiver. Fig. 3(c) shows an example of a cluster tree topology of ZigBee for two-level clustering. In the WISN, BS is usually resource-rich device such as a PDA, laptop and desktop computer equipped with a sensor mote and connects to the Internet or intranet. Thus, BS can communicate with the server and controller devices via wire or wireless technologies. In industrial environments, a hierarchical architecture often utilizes heterogeneity of motes [19]. Although we consider homogeneous WISN, our framework can adopt heterogeneity for two-level clustering. Fig. 3(d) shows an example of two-level clustering based on heterogeneity using gateway motes that can communicate via a robust medium, such as the MIB family and Stargate NetBridge [41]. These motes connected to sensor motes can be used as a bridge to link the WISN and other wireless networks, as well as wired networks in industrial environments.

In the second and the third cases considering deployment errors, more sophisticated clustering algorithms is necessary for achieving two-level clustering. In these cases, as mentioned above, the existing cluster algorithms can be utilized according to applications of WISNs, since clusters are formed dynamically after deployment in those algorithms. For example, Chen *et al.* proposed an Evenly Distributed Clustering (EDC) algorithm to form $K$-hop clustering [7]. In EDC, each node is at most $K$-hops away from a cluster head and $K$ can be determined according to the requirements of applications. It also can minimize the number of $K$-hop clusters and evenly distribute clusters across the sensing field. In our two-level clustering, $K$ can be 1 for one-hop clustering, while $K$ can be more than 2 for multihop clustering simultaneously. Although the EDC algorithm is not specialized algorithm for mobile nodes, several clustering algorithms [13] proposed for mobile sensor networks may be employed for industrial applications using mobile sensors.

### B. Intrusion Detection Based on Two-Level Clustering

Due to the inherent properties of WISNs, such as the deployment in inaccessible and harsh environments and the broadcast and wireless nature of transmission, there are several difficulties in managing WISNs effectively and detecting various types of adversarial intrusions over wireless medium. First, WISNs can be easily exposed to various attacks such as eavesdropping, illegal modification, and fabrication due to the use of wireless medium. In addition, it is not easy to detect various types of adversarial intrusions over wireless medium without monitoring all communication traffics as well as sensor nodes, which is quite impractical. Especially, it is more difficult to detect inside attacks in which attackers redistribute the compromised nodes to the network for further attacks using the capability of stealing the key materials contained within them after capturing and compromising some sensor nodes physically. Those adversarial intrusions may result in more severe disasters around industrial environments. Thus, it is inevitable that WISNs must require intrusion detection mechanisms which detect such attacks and intrusions, and alert the network for considering countermeasures. For this purpose, we propose an intrusion detection protocol based on the above mentioned two-level clustering.

*1) Basic Detection Techniques:* Like as the existing IDSs in WSNs, we utilize the watchdog technique. The detailed detection techniques against the well-known attacks will be presented on the basis of experiments in Section IV-B. Industrial applications can apply the detection techniques from the results of experiments selectively to the module of intrusion detection rules according to their different requirements and the capability of sensor nodes. Our intrusion detection protocol can also be easy to add detection techniques of new kinds of attacks for the future without the modification of the proposed protocol.

*2) Hierarchical Intrusion Detection Protocol:* Fig. 4 shows sensor nodes' modules and the flow chart for processing intrusions according to a role: a gateway (GW), cluster head (CH) or member node (MN). All nodes have two basic modules: *Event Sensing* and *Data Aggregation*. A MN simply delivers the sensed data to its higher level (a CH). Each GW and CH aggregate and process the data delivered from the lower levels (CHs or MNs, respectively) and then transmit it to a higher level (the BS or GW, respectively). Each node also has a IDS module. IDS module has two submodules: *Intrusion Detection Rules* that decides an intrusion through applying detection rules and threshold to the neighbor's traffic and *Intruder Handling* that reactively handles the intruder. As mentioned above, each industrial application can employ different detection techniques to the module of intrusion detection rules according to its security requirements. Once a condition of rules in the module of intrusion detection rules is satisfied, a sensor node concludes that a malicious intrusion occurs around it and reactively handles the intrusion depending on the module of intruder handling. The details are described in the following subsections.

Intrusion detection within each level and between levels operates by eavesdropping traffic within one-hop, and by evaluating the transmitted control and sensing messages. As we mentioned, two-level clustering generates four levels: base station (BS),

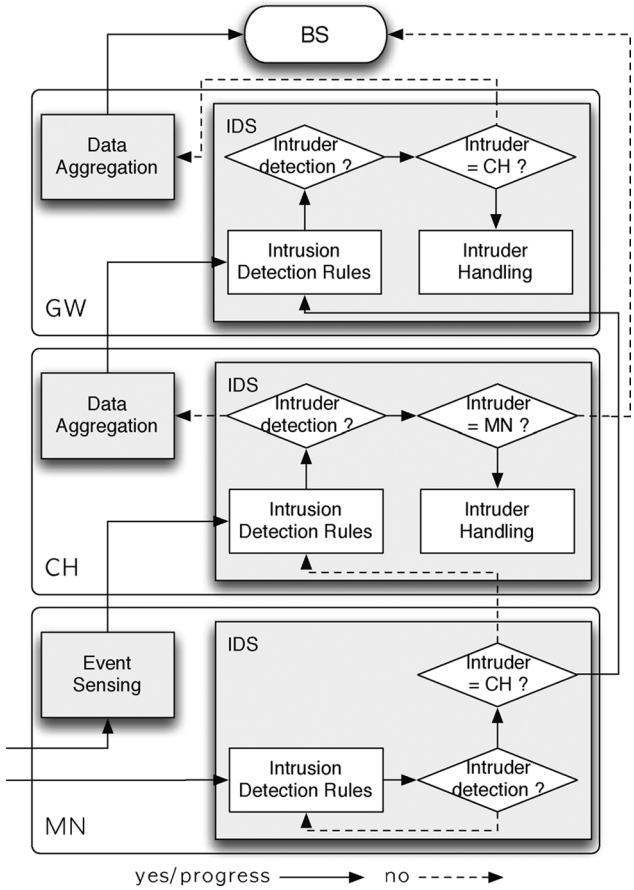


Fig. 4. Basic modules according to the roles of sensor nodes and the flow chart of intrusion detection in the hierarchical framework.

gateway (GW), cluster head (CH) and member node (MN). Although each level detects intrusions with the similar detection rules, each level performs a different handling method. If a MN detects an intruder among its neighbors including a GW and CH, it does not handle it by itself and only reports it to a higher level. A CH monitors a MN and GW within its communication range (one-hop) directly while the CH indirectly monitors a GW located at the outside of its communication range through evaluating messages that the GW transmits. If the CH detects a malicious MN as an intruder, it deals with the intruder (i.e., the CH removes the intruder from its cluster). On the other hand, if it detects a malicious GW as an intruder, the CH reports it to BS, the root level. A GW monitors own member CHs. When the GW detects a malicious CH as an intruder, it removes the CH from its cluster and reports it to the BS. When a CH or GW is reported as an intruder, the BS instructs the network to re-perform the second clustering or the first clustering, respectively.

a) Monitoring member nodes.
Each MN senses environmental events and transmits them to its CH. The MN also plays a role as an intermediate node which relays a packet. In the second clustering, A CH is able to detect a malicious MN since all MNs are located within CH's communication range (one-hop). Once the CH detects the malicious MN, it increases its abnormal counter. If the abnormal counter for the malicious MN is above the predetermined threshold, then this node

is considered as an intruder and it is removed by the CH from the network. When a malicious MN plays a role an intermediate node, its CH may not detect its misbehavior. Thus, it is desirable for each MN to monitor intermediate nodes among its neighbors even though they belong to other clusters. If the MN detects their misbehavior, the MN reports it to its CH. If the CH receives a report that its MN is an intruder, it deals with the corresponding MN. On the other hand, if the CH receives a report that other clusters' MN is an intruder, it informs to the corresponding CH.

b) Monitoring cluster heads.
After the second clustering, it is natural that each MN monitors own CH since they can mutually overhear their network traffic. For monitoring a CH, its MNs are divided into groups according to a sleep/wake schedule. MNs of each group then collaborate with each other to monitor its CH. Once a MN detects the misbehavior of the CH, it increases the abnormal counter of the CH and updates this counter by sharing it with its neighbors belong to the same cluster. If the abnormal counter of the CH is greater than the predetermined threshold, the MN reports to GW or other CH over the paths where the malicious CH does not include. In addition, GWs have the advantage to detect the misbehavior of their CHs since every CH sends the gathered data to its GW. Once the GW detects the misbehavior of a CH or receives the reports from its MNs, it restrains the malicious CH and reports it to the BS. The BS then instructs to re-perform the second clustering.

c) Monitoring gateways.
Each GW plays important roles that it processes the gathered data from its CHs and sends finally the processed data to the BS. Thus, it is important to detect a malicious GW. Both all nodes including CHs within a GW's communication range (one-hop) and CHs which are not located within its communication range, but are its member CHs, should monitor the GW and detect its misbehavior. When a MN detects a malicious GW, it reports it to its CH. On the other hand, a CH reports to the BS when it receives the report it from MNs or detects the malicious GW by itself. When the BS ultimately determines the GW is an intruder, it removes the intruder from the network and instructs to re-perform the first clustering.

### C. Intrusion Prevention Based on Two-Level Clustering

In industrial environments, critical information, such as proprietary algorithms and data, should be kept secret from competitors or customers [9]. Moreover, there can be remained several attacks of which detection techniques are not known, such as an eavesdropping, bogus routing information attack and sink hole attack. In general, cryptographic mechanisms such as key management and authentication protocols can be used to prevent various intrusions prior to intrusion detection. We introduce an intrusion prevention protocol in which mutual authentication is adopted for nodes at the different levels using a key establishment protocol in WSNs.

*1) Key Establishment:* Message encryption and authentication based on key management protocols are needed to secure WISNs against malicious access. As we mentioned in Section III-A, we can consider at least three cases of deployment with regard to WISNs. In the first case, without the need to consider any key management protocols, we can load keys for each node before its deployment with careful deployment plan and construction. On the other hand, we can utilize deterministic key protocols for WSNs due to possible obstructions and deployment errors in the last two cases.

We especially utilize Localized Encryption and Authentication Protocol (LEAP) [39] which is suitable for the last two cases since a node can dynamically establish pairwise keys with any immediate neighbors using an initial key regardless of possible deployment errors and it supports easy addition of new node. Furthermore, it gains advantages over probabilistic schemes in memory, communication and computational overheads, as well as global connectivity which is always 1. We establish individual keys $(K_{BS,GW}, K_{BS,CH}, K_{BS,MN})$ shared with BS and pairwise keys $(K_{CH,MN}, K_{MN,MN})$ shared with neighbors including a CH within own communication range (within one-hop), using LEAP. In this paper, the shared keys $(K_{GW,CH})$ between a CH and GW, located at more than two-hop away from each other, are established by a path key establishment phase [10]. Each cluster key $(K_C)$ shared between a CH and its all MNs are used only in the second clustering. The cluster key is generated by the CH and forwarded one-to-one from the CH to its MNs in the secure manner (i.e., by encrypting it with a pairwise key).

*2) Authentication Protocol for Intrusion Prevention:* In WISNs, messages can be divided largely into two types: control messages transmitted from the BS to the lower levels and sensing messages transmitted from MNs to the higher levels and to the BS in the end. Each CH processes the sensed data from its MNs and sends it to its GW. Similarly, each GW gathers and processes data received from its member CHs, and sends it to the BS. Sensing messages in the second clustering are transmitted within one-hop (within the CH's communication range). In this case, it is easy for each CH to detect an adversary that inserts false data into a packet or alters data of a packet in the second clustering. However, the adversary can insert fault data or alter data in the multihop communication (i.e., communication between a CH and a GW or between a GW and the BS). In this case, it is difficult for CHs to determine whether the packet is transmitted from the adversary or one of MNs, legitimate nodes. Therefore, it is necessary to authenticate both data and nodes.

In this paper, all sensing messages are appended with Message Authentication Code (MAC) to provide data and node authentication. Furthermore, partial sensing messages are able to encrypt according to industrial applications and the importance of their data that should be kept in secret in order to prevent eavesdropping and to provide confidentiality. Control messages can be transmitted to a specific node, to a specific cluster or to the whole network. In the first two cases, the BS, GW, and CH can be authenticated using multihop authentication (i.e., generation and verification MAC with keys between nodes of the different hop). Like as the case of sensing messages, some control messages can also encrypt according to the applications and importance of data (i.e., rekeying for backward and forward se-

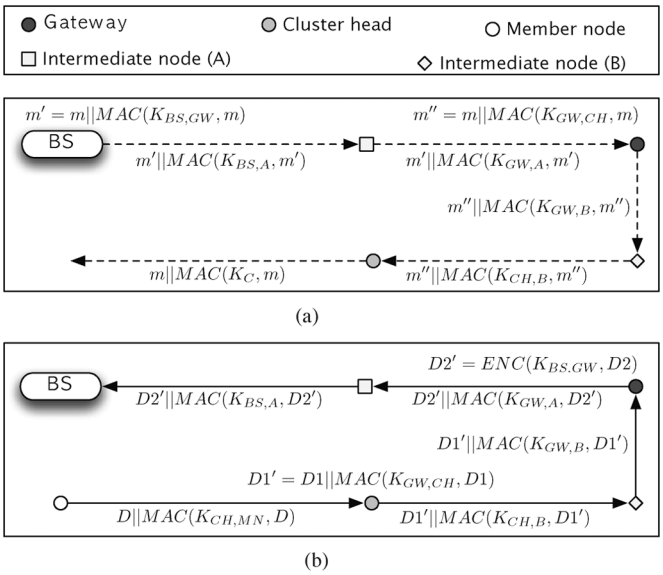


Fig. 5. A hierarchical intrusion prevention protocol. (a) Control message authentication. (b) Sensing message authentication.

crecy). In the latest case, it is possible for all nodes to authenticate the BS utilizing $\mu$-TESLA [24]. Since encryption of all data may be expensive in terms of computation, we consider only authentication and partial encryption of sensing messages. Fig. 5(a) and (b) show examples of authentication and partial encryption of a control message and sensing message, respectively. In Fig. 5, a control message or sensing message $m$ is authenticated in a hop-by-hop fashion through $MAC$ with pairwise keys. The sensing message transmitted from the GW to BS is encrypted with the corresponding individual key. In Fig. 5, we use the following notations.

- $a||b$: a concatenation of $a$ and $b$ strings;
- $K_{x,y}$: a shared key between nodes $x$ and $y$;
- $\mathrm{ENC}(K, m)$: encryption of a message $m$ with a key $K$;
- $\mathrm{MAC}(K, m)$: a message authentication code of a message $m$ with a key $K$

Our intrusion prevention protocol does not impose restrictions on symmetric cryptography algorithms for encryption/decryption and computing/verifying MACs. Since a timely data transmission is a critical issue in industrial environments, computational overheads of those algorithms must not corrupt the other tasks of applications regarding real time performance. The various results of analyzing computational overhead of those algorithms have been presented in the literature [17], [40], [11]. They show that symmetric algorithms are feasible for WISNs in which impose real-time constraints. For example, TinySec uses CBC (cipher block chaining) for encryption and CBC-MAC for message integrity. In [17], time to execute RC5 and Skipjack, 64 bit block cipher, on the MICA2 sensor nodes are 0.90 ms and 0.38 ms, respectively. The CC2420 provides IEEE 802.15.4 medium access control hardware security operations [40]. This includes AES-CTR (counter mode) encryption/decryption, AES-CBC-MAC authentication, AES-CCM authentication and encryption, and the stand-alone AES encryption. Table I describes security timing examples

TABLE I
SECURITY TIMING EXAMPLES OF CC2420 [40]

| Mode | l(a) | l(m) | l(mic) | Time($\mu s$) |
|---|---|---|---|---|
| CCM | 50 | 69 | 8 | 222 |
| CTR | - | 15 | - | 99 |
| CBC | 17 | 98 | 12 | 99 |
| Stand-alone | - | 16 | - | 14 |

**[Notation]** **l(x)**: byte length of x, **a**: authentication payload, **m**: message, and **mic**: message integrity code

TABLE II
POSSIBILITIES OF INTRUSION PREVENTION AND DETECTION BASED ON THE PROPOSED PROTOCOLS

| Attack | Attacker | Extent | P | D |
|---|---|---|---|---|
| Eavesdropping | S/M | L/W | △ | × |
| Spoofed, altered, replayed packet attacks | S | L | △ | ◯ |
| Bogus routing info. attack | S | L | ◯ | • |
| Hello Flooding attack | S/M | L/W | ◯ | • |
| Selective forwarding attack | S | L/W | × | ◯ |
| Sybil | S/M | L/W | △ | ◯ |
| Sinkhole attack | S/M | L/W | △ | ◯ |
| Packet jamming attack | S/M | L/W | × | ◯ |
| RF jamming attack | S | L/W | × | △ |

**Attacker**: the number of attackers, S: a single attacker, M: multiple attackers,
**Extent**: the extent of attacks, L: the local network, W: the whole network,
**P**: the possibility of intrusion prevention, **D**: the possibility of intrusion detection,
×: impossibility, △: partial possibility, ◯: good possibility,
•: unnecessariness of intrusion detection

of CC2420. It shows that time used by the security module of CC2420 for IEEE 802.15.4 medium access control encryption and authentication (AES-CCM) is at most 222 $\mu$s. Furthermore, P. Ganesan *et al.* [11] evaluated the feasibility of popular symmetric cryptography algorithms for a range of embedded architectures used by practical sensor motes, such as Atemega103, Atemega128, StrongARM, XScale. For this purpose, they presented computational overheads of those algorithms (RC4, IDEA, MD5, SHA-1, RC5), regarding time to execute the operations, the related clock cycles and so on. In conclusion, most of symmetric cryptography operations are reasonable for WISNs of real-time industrial applications. According to their security and performance requirements, industrial applications can employ them for our intrusion prevention protocol.

Our intrusion prevention protocol enables to protect the network prior to intrusions and our intrusion detection protocol enables to secure the network against intrusions that cannot be defended by the prevention protocol. Table II shows whether intrusions can be prevented, detected or not when the proposed authentication protocol and intrusion detection protocol are used in the same time. For example, it is unnecessary to detect a bogus routing information attack and hello flooding attack (• in **D** in Table II) because they can be prevented through the intrusion prevention protocol. In general, an eavesdropping attack can be prevented through encryption. However, the intrusion detection protocol adopts eavesdropping in order to monitor traffic of neighbors (× in **D**) and only the sensing message between the BS and GW is encrypted in our framework (△ in **P**).

## IV. EXPERIMENTAL RESULTS

In this section, we conduct experiments on the previous IDSs devised for WSNs [28], [2], [23], [15], [31], [26], we call these

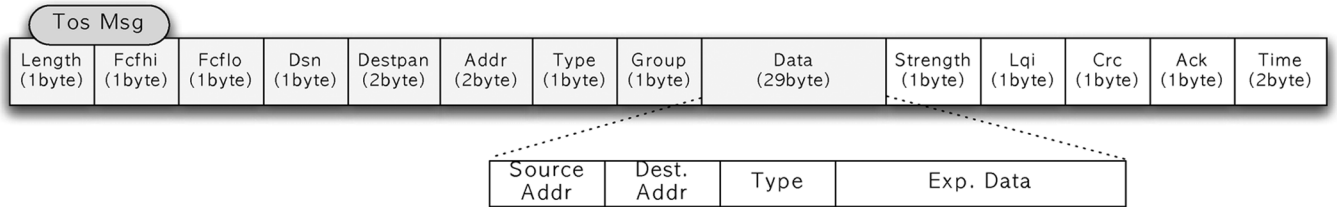


Fig. 6. The packet format for TinyOS 2.0.

experiments *Detection technique test*. Through those experiments with real motes (i.e., MICAz and Telos [41]), we compare detection techniques proposed in the previous IDSs and select better techniques against various kinds of attacks that are harmful to WISNs. Based on the experiments, we yield novel results on the previous techniques related to specific attacks. We also test the performance of two detection techniques to show the feasibility of them in real-time WISNs. We conduct another experiment on a detection technique with real motes to show that one-hop clustering is necessary for intrusion detection and we call these experiments *one-hop versus multihop test*.

### A. Environment of Experiments

The environment for experiments is as follows. In detection technique test, we use 12 MICAz motes and 4 Telos rev.B motes [41] used as sensor nodes in practice. In one-hop versus multihop test, we use 16 MICAz motes for member nodes, and two Telos rev. B motes for a cluster head and malicious head. We use three portable PCs connected with Telos rev. B; one for the BS and the other for saving the result of detection. Their operating system is Microsoft Windows XP and we use JAVA programming language to save the results.

MICAz and Telos rev.B are ZigBee-ready wireless SmartDust sensors. Both Telos and MICAz platforms use the CC2420 radio stack, a true single-chip 2.4 GHz IEEE 802.15.4 compliant RF transceiver, and TinyOS [44] which is a event-driven OS for WSNs. CC2420 adopts the IEEE 802.15.4 standard, while it does not adopt the network/application layers defined by the ZigBee's network/application layers. Although a ZigBee protocol could be written in TinyOS, we use the simple flooding supported by TinyOS for routing instead of routing protocols defined by ZigBee. All attacks and detection techniques are implemented by NesC [43] in TinyOS. NesC is a compiler for a new language designed to support TinyOS. The format of all the packets for experiments follows the frame format for IEEE 802.15.4 packets in TinyOS 2.0 in Fig. 6. The first six bytes from "Length" to "Destpan" in Fig. 6 are reserved for handling ZigBee messages. Since CC2420 does not follow the ZigBee network/application layers, we put all data (source address, destination address, message type and experimental data) needed for our experiments in the 29-byte "Data" section (see Fig. 6).

### B. Detection Technique Test

In order to compare original efficiency of detection techniques, we consider the same network configuration with a flat structure, instead of a hierarchical structure. In the flat structures, each node typically plays the same role as a monitoring node. We also consider general environments for various industrial applications, instead of considering specific factory environments. As we mentioned, WISNs are vulnerable to various types of attacks that are also configurable against WSNs. In our view, technical details of such attacks can be generalized with regard to WSNs, for example, node capture, eavesdropping, impersonation, and infiltration over wireless channels. Thus, we need to consider various types of attacks on WSNs and the corresponding detection techniques. We conduct experiments on the representative attacks in WSNs and the corresponding detection techniques from the previous IDSs studied for WSNs. The goal of these experiments is to choose the best intrusion detection technique for WISNs from the existing IDSs of WSNs.

In general, industrial applications may require both reliable and real-time communications. For reliable communications, the proposed intrusion detection and prevention protocols can be used. However, they must not corrupt real-time performance of industrial applications. In Section III-C2, we already mentioned that symmetric cryptography operations implemented on hardware are only used for intrusion prevention and they may have negligible influence on the original performance. In addition, the continuous check of the intrusion detection rules using the detection techniques must not influence the real-time performance as well. For showing this, we additionally conduct an experiment to evaluate the amount of time required on real motes in Section IV-B3.

As for the bandwidth usage, there is no additional packet transmission required for our scheme unless any intrusion is detected. Note that if any intrusion is detected, the management of discovered intrusion must have the highest priority. The influence on the performance resulting from this task is less important.

Through two kinds of experiments, our hierarchical intrusion detection protocol enables industrial applications to employ several or all detection techniques against different types of attacks according to their security and performance requirements.

*1) Intrusion Detection Techniques Against Well-Known Attacks:* We conduct experiments on various attacks, such as eavesdropping, routing attacks [16] and DoS attacks [36], as well as the corresponding detection techniques extracted from IDSs of WSNs. These attacks and detection techniques are the followings.

- Packet jamming attack (DoS attack): This attack interferes with the radio frequencies through sending a lot of packets repeatedly. We consider two types of attackers; one mote attacker (PA1) and several mote attackers (PA2).

— Detection using the packet arrival rate (PD1) [23].
— Detection using the predetermined packet arrival threshold (PD2) [31].

- Impersonation attack [23]: This attack disrupts the network through impersonating one of the legitimate node by spoofing the ID. We consider two types of attackers; the first attacker impersonates one of neighbors by spoofing its ID (IA1) and the second attacker impersonates one of nodes removed from the network or nonexisting node using random ID (IA2).
  — Detection using the so-called "Radio transmission range" rule (ID1) [28].
  — Detection using the point that a node can easily detect the suspicious node that uses the destroyed nodes (ID2) [23].
- Hello flooding attack (routing attack): In this attack, an attacker tricks a lot of nodes by flooding hello messages with a high-powered transmitter into believing the nodes are neighbors of the attacker.
  — Detection using an alarm about new node addition (HD1) [26].
- Spoofed/altered packet attack: This attack spoofs or alters a packet in order to make industrial applications or devices malfunction.
  — Detection using the so-called "Integrity rule." (AD1) [28]
  — Detection using an anomaly detection table (ADT) which contains the list of neighbors that may forward some particular information to that node (AD2) [2].
  — Detection using the point that all data follow certain patterns and limits (AD3) [26].
- Selective forwarding attack/Packet dropping attack (routing attack): This attack occurs when a compromised node forwards certain packets selectively and drop others simply.
  — Detection using the so-called "Interval rule" (SD1) [28].
  — Detection using a predefined specification for the selective forwarding attack (SD2) [15].
  — Detection by checking packet dropping (SD3) [31].

*2) Results of Detection Technique Test:* Table III shows the results of detection technique test. In this test, there was no false alarm and the missed detection rate for each attack is all but the detection rate. The missed detection of the altered packet attack and selective forwarding attack includes a case that nodes located outside of the attacker's transmission range cannot detect those attacks at all. Through the results of experiments, we select better detection techniques and disclose several unknown features of the previous detection techniques related to specific attacks in the following.

- Packet jamming attack: Table III shows that PD1 is suitable for PA1, while PD2 is suitable for PA2. PD1 or PD2 can be selected according to applications. In particular, PD1 is more suitable for industrial applications in which the packet arrival rate is needed to compute in real-time. Through the experiments, we find out that this attack can actually drop packets as an RF jamming attack does rather than flooding an enormous number of packets onto nodes nearby. It should imply that a detection technique against this attack has to detect an abrupt decrease of receiving rate as well as an increase of it sensitively. However, the previous techniques only consider such as increase beyond the average packet arrival rate or predetermined threshold. Especially, it is important for WISNs to detect a packet jamming attack as RF noise since abundant RF noise is a critical concern in industrial environments [19]. Thus, we allow each node to detect the packet jamming attack when the average packet arrival rate decreases as well as increases remarkably. With this detection technique, we obtain detection rate of 80.4%, 86.4%, 72.7%, 84.0% for PD1 and PD2 against PA1 and for PD1 and PD2 against PA2, respectively.
- Impersonation attack: ID1 is more suitable for detecting both IA1 and IA2. In result, it is possible to use only the neighbor list for detecting IA1 and IA2 of the outside attacker. However, it is difficult for ID1 to detect an insider attacker which impersonates a legitimate node. To solving this problem, we allow each node to check whether neighbors impersonate as own ID by monitoring all traffic within own communication range.
- Hello flooding attack: In HD1, each node sounds an alarm about a new node addition and checks whether the new node is actually added through a query to the BS. Although HD1 has the high detection rate as 94.9%, there may be high communication overheads with respect to transmitting queries and replies. Communication overheads will grow when the adversary tries to launch the hello flooding attack to the wide area with the stronger signal. To solve this problem, we allow the BS to forewarn the network of new node addition by authenticated broadcast mechanisms (i.e., $\mu$TESLA). If a GW and CH receive a hello message, they can decide to accept or reject the hello message through the BS's broadcast message.
- Spoofed/altered packet attack: AD2 and AD3 have the same detection rate, but AD2 has an additional overhead for keeping ADT. In general, a CH and GW can ignore the

TABLE III
AVERAGE DETECTION RATE ACCORDING TO ATTACKS USING INTRUSION DETECTION TECHNIQUES

| Attacks | | Detection techniques | Detection rate(%) |
|---|---|---|---|
| Packet jamming attack | PA1 | PD1 [23] | ≈67.5 |
| | | PD2 [31] | ≈53.9 |
| | PA2 | PD1 [23] | ≈45.8 |
| | | PD2 [31] | ≈67.7 |
| Impersonation attack | IA1 | ID1 [28] | 50.9 |
| | | ID2 [23] | 0 |
| | IA2 | ID1 [28] | 100 |
| | | ID2 [23] | 60 |
| Hello flooding attack | | HD1 [26] | 94.9 |
| Spoofed/altered packet attack | | AD1 [28] | 25 |
| | | AD2 [2] | 42.5 |
| | | AD3 [26] | 42.5 |
| Selective forwarding attack | | SD1 [28] | 9.9 |
| | | SD2 [15] | 9.9 |
| | | SD3 [31] | 26.1 |
| Eavesdropping | | non-exist | x |
| Bogus routing info. attack | | non-exist | x |
| Sinkhole attack | | non-exist | x |

sensed data which is completely different data from others since they have the task of data aggregation and processing in the clustered network. However, it is difficult to know exactly who the intruder is among MNs if they are far from each other. In this case, the help of MNs is needed. For detecting this attack, each node can save the packets of neighbors temporarily and check whether neighbors transmit the same data, like as AD1. Altered or false data from WISNs can cause damage to industrial applications or even have a dangerous effect on industrial environments [19]. Thus, both AD3 and AD1 should be used to detect this kind of attacks. As the above mentioned, AD3 may be performed naturally in the clustered network. Although AD1 has a lower detection rate than AD2 and AD3, it is plausible that we employ AD1 for detecting this attack since it can be used with SD3 with regard to a similarity between their techniques. In AD 1 and SD3, each node stores the packets of neighbors to detect an altered packet attack and selective forwarding attack.

- Selective forwarding attack (packet dropping attack): SD1 and SD2 have very low detection rate since only MNs which monitor both receiving and sending messages of the attacker can detect this attack. SD3 is more suitable for detecting this attack due to the higher detection rate. In industrial environments, WISNs should be more robust to this kind of attacks than WSNs with regard to reliability and real-time communication as the typical requirements [14] for WISNs. Thus, SD3 can be useful for detecting this attack. As the mentioned above, SD3 can be used with AD1 since each node saves the packets of neighbors for a while to check whether neighbors actually transmit or drop them.
- Eavesdropping, bogus routing information attack and sinkhole attack: Detection techniques of these attacks are not known. Especially, since eavesdropping is utilized as the watchdog technique for intrusion detection, it is impossible to detect this attack. In industrial applications, there is information that should not be disclosed to attackers, including proprietary algorithms or data [14]. Our intrusion prevention protocol in Section III-C2 can be useful to protect those data from eavesdroppers. A bogus routing information attack and sinkhole attack are originated from updating routing, so they may be prevented by node authentication in Section III-C2.

Through these experiments, we obtain better detection techniques against possible attacks on WISNs.

*3) Results of Real-Time Performance Test:* We conduct experiments additionally on two detection techniques (AD3 and ID1) using Telos motes with respect to the execution time. In this test, we evaluate time to check the intrusion detection rules with AD3 and ID1 continuously. Over 100 times tests per each technique, it takes each node the average 20.1 and 27.5 $\mu$s to check AD3 and ID1 per one packet, respectively. This indicates that checking the detection rules may not have much effect on real-time tasks of industrial applications. Based on the above results, our hierarchical intrusion detection protocol enables industrial applications to employ several or all detection techniques against different types of attacks considering their security and performance requirements.

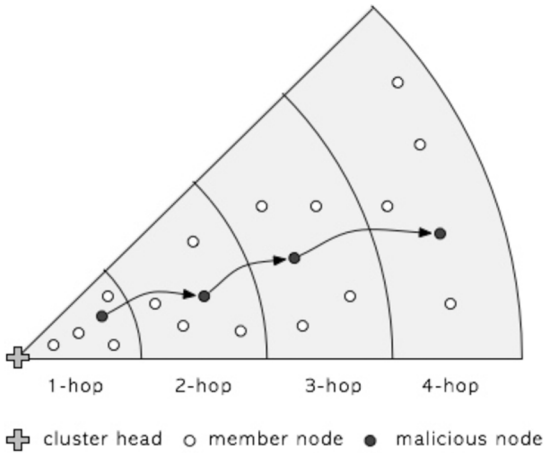


Fig. 7. Network configurations for one-hop versus multihop test.

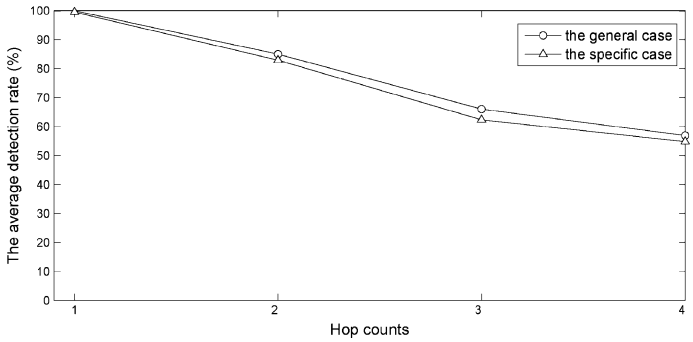


Fig. 8. The average detection rate of a packet spoofing attack according to hop counts in the one-hop versus multihop test, where $s = e = 0.1$.

### C. One-Hop Versus Multihop Test

We conduct experiments on the detection rate according to the distance of a malicious node from a cluster head. The goal of this experiment is to show that one-hop clustering is essential for intrusion detection, so that a cluster head detects malicious nodes as intruders within its cluster using direct monitoring. In this test, we consider two cases of a four-hop cluster in the example in Section II-B2; the general case regardless of the specific detection technique and attack and the specific case regarding of a specific detection technique (i.e., a spoofed packet attack) to sensor nodes including the cluster head against the corresponding attack. In the general case, the detection rate of the cluster head is evaluated by the arrival rate of packets sent periodically from member nodes. Namely, the detection rate of the general case shows how the reliability of member nodes influences the normal transmission rate according to the distance of nodes from a cluster head. In the specific case, the detection rate is regarded as the real results of detecting one malicious node which launches the spoofed packet attack. For this test, we intentionally deploy nodes including a cluster head and malicious node instead of using a clustering algorithm and we change one malicious node's location as following the arrow in Fig. 7. We use one Telos mote for the cluster head, 16 MICAz motes for member nodes including intermediates and one laptop PC for saving the detection results from the cluster head. We deploy 1 cluster head in the center of the four-hop cluster and four

member nodes are deployed at every distance one-hop, two-hop, three-hop and four-hop from the cluster head. We assume that $s$ and $e$ of the example are 0.1, respectively, so that we make a member node to drop 10% packets randomly and put into the sleep state periodically.

Fig. 8 shows the detection rate for two cases according to the distance between the cluster head and malicious node. In this test with real motes, the additional packet loss from packet collision causes the gap between the average detection probabilities in the case of $s = e = 0.1$ in Fig. 2 and the detection rate in the general case in Fig. 8. Like the graphs in Fig. 2, the graphs of detection rate for both cases fall linearly according to increasing the distance between the malicious node and cluster head.

Thus, one-hop clustering is necessary for intrusion detection in the hierarchical networks including WISNs since the reliability of nodes, especially intermediates, has influence on intrusion detection. Since the proposed intrusion detection protocol uses one-hop clustering for intrusion detection, the cluster head can detect intrusions directly with regardless of the reliability of nodes such as their sleep rate and error rate.

## V. Conclusion

We first study intrusion detection for WISNs through various experiments on the previous IDSs devised for WSNs, with real motes. We classify better methodologies against various kinds of attacks and yield several novel results on the previous methodologies. We stress on the significance of one-hop clustering for intrusion detection. We propose a hierarchical framework in the way of considering both intrusion detection and data processing and construct hierarchical intrusion prevention and detection protocols in the framework.

We believe that our hierarchical framework is useful for securing industrial applications with regard to two lines of defense. We also believe that our framework makes it easy to apply new detection techniques against further attacks to the intrusion detection protocol.

In the future study, we expect that the proposed framework will be investigated for heterogeneous WISNs. Furthermore, we expect that we propose a new suitable clustering algorithm for our framework.

## References


[1] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proc. INFOCOM'03*, 2003, pp. 1713–1723.

[2] V. Bhuse and A. Gupta, "Anomaly intrusion detection in wireless sensor networks," *J. High Speed Networks*, vol. 15, pp. 33–51, 2006.

[3] A. Bonivento, C. Fischione, L. Necchi, F. Pianegiani, and A. Sangiovanni-Vincentelli, "System level design for clustered wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 3, no. 3, pp. 202–214, Aug. 2007.

[4] P. Brutch and C. Ko, "Challenges in intrusion detection for wireless ad-hoc networks," in *Proc. Appl. Internet Workshops*, 2003, pp. 368–373.

[5] E. T. Capo-Chichi, H. Gutennet, and J.-M. Friedt, "IEEE 802.15.4 performance on a hierarchical hybrid sensor network platform," in *Proc. IEEE 5th Intl. Conf. Networking and Services*, 2009, pp. 303–308.

[6] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proc. IEEE Symp. Res. Security and Privacy*, 2003, pp. 197–213.

[7] Q. Chen, J. Ma, Y. Zhu, D. Zhang, and L. M. Ni, "An energy-efficient k-hop clustering framework for wireless sensor networks," in *Proc. EWSN'07*, 2007, vol. 4373, LNCS, pp. 17–33.

[8] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key predistribution scheme for wireless sensor networks," in *Proc. ACM CCS'03*, 2003, pp. 228–258.

[9] D. Dzung, M. Naedele, T. P. Von Hoff, and M. Crevatin, "Security for industrial communication systems," in *Proc. IEEE*, 2005, vol. 93, no. 6, pp. 1152–1177.

[10] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. 9th ACM Conf. Comput. Commun. Security*, Nov. 2002, pp. 41–47.

[11] P. Ganesan, R. Venugopalan, P. Peddabachagari, A. Dean, F. Mueller, and M. Sichitiu, "Analyzing and modeling encryption overhead for sensor network nodes," in *Proc. WSNA'03*, Sep. 2003, pp. 151–159.

[12] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proc. 5th ACM/IEEE Mobicom*, Aug. 1999, pp. 174–185.

[13] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Hawaii Int. Conf. Syst. Sciences*, Jan. 2000, pp. 3005–3014.

[14] J. Heo, J. Hong, and Y. Cho, "EARQ: Energy aware routing for real-time and reliable communication in wireless industrial sensor networks," *IEEE Trans. Ind. Informat.*, vol. 5, no. 1, pp. 3–11, Feb. 2009.

[15] K. Ioannis and T. Dimitriou, "Towards intrusion detection in wireless sensor networks," in *Proc. 13th Eur. Wireless Conf.*, 2007.

[16] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in *Proc. 1st IEEE Int. Workshop on Sensor Network Protocols and Appl.*, May 2003, pp. 113–127.

[17] C. Karlof, N. Sastry, and D. Wagner, "TinySec: A link layer security architecture for wireless sensor networks," in *Proc. Sensys'04, ACM*, Nov. 2004, pp. 162–175.

[18] H. Körber, H. Watter, and G. Scholl, "Modular wireless real-time sensor/actuator network for factory automation applications," *IEEE Trans. Ind. Informat.*, vol. 3, no. 2, pp. 111–119, May 2007.

[19] L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Dushalnagar, L. Nachman, and M. Yarvis, "Design and deployment of industrial sensor networks: Experiences from a semiconductor plant and the north sea," in *Proc. Sensys'05*, 2005, pp. 64–75.

[20] T. Kwon and S.-H. Park, "Experimental study on wireless sensor network security," in *Proc. ISI 2006*, 2006, vol. 3975, LNCS, pp. 741–743.

[21] C. E. Loo, M. Y. Ng, C. Leckie, and M. Palaniswami, "Intrusion detection for routing attacks in sensor networks," *Int. J. Distrib. Sens. Netw.*, vol. 2, no. 4, pp. 313–332, Dec. 2006.

[22] K. S. Low, W. N. N. Win, and M. J. Er, "Wireless sensor networks for industrial environments," in *Proc. IEEE CIMCA-IAWTIC'05*, 2005, pp. 271–276.

[23] I. Onat and A. Miri, "An intrusion detection system for wireless sensor networks," in *Proc. IEEE Int. Conf. Wireless and Mobile Comput., Netw. Commun. (WiMobapos 2005)*, Aug. 2005, vol. 3, pp. 253–259.

[24] A. Perrig, R. Szewczyk, W. Wen, D. Culler, and J. D. Tygar, "SPINS: Security protocols for sensor networks," *Wireless Netw.*, pp. 521–534, 2002.

[25] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, "Distributed anomaly detection in wireless sensor networks," in *Proc. 10th IEEE Singapore Int. Conf., Commun. Syst.*, 2006, pp. 1–5.

[26] R. Roman, J. Zhou, and J. Lopez, "Applying intrusion detection systems to wireless sensor networks," in *Proc. IEEE Consumer Commun. Netw. Conf.*, Jan. 2006, pp. 640–644.

[27] X. Shen, Z. Wang, and Y. Sun, "Wireless sensor networks for industrial applications," in *Proc. IEEE WCICA 2004*, 2004, pp. 3636–3640.

[28] A. da Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Proc. 1st ACM Int. Workshop on Quality of Service Security in Wireless and Mobile Networks*, 2005, pp. 16–23.

[29] B. Sinopoli, C. Sharp, S. Schaffert, and S. S. Sastry, "Distributed control applications within sensor networks," in *Proc. IEEE*, Aug. 2003, vol. 91, no. 8, pp. 1234–1246.

[30] A. A. Strikos, "A full approach for intrusion detection in wireless sensor networks," *School of Information and Communication Technology*, Mar. 2007, KTH.

[31] C.-C. Su, K.-M. Chang, Y.-H. Kuo, and M.-F. Horng, "The new intrusion prevention and detection approaches for clustering-based sensor networks," in *Proc. IEEE WCNC'05: Broadband Wireless for the Masses Ready for Takeoff*, Mar. 13–17, 2005, pp. 1927–1932.

[32] K. Sun, P. Peng, P. Ning, and C. Wang, "Secure distributed cluster formation in wireless sensor networks," in *Proc. 22nd ACSAC'06*, Dec. 2006, pp. 131–140.

[33] J.-P. Thomesse, "The WorldFIP fieldbus," in *The Industrial Information Technology Handbook*, R. Zurawski, Ed. Boca Raton, FL: CRC, 2005.
[34] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Commun. Surveys Tutorials*, pp. 2–23, 2006.
[35] A. Willig, "Recent and emerging topics in wireless industrial communications: A selection," *IEEE Trans. Ind. Informat.*, vol. 4, no. 2, pp. 102–124, May 2008.
[36] A. D. Wood and J. A. Stankovic, "Denial of service in sensor network," *IEEE Computer*, pp. 54–62, Oct. 2002.
[37] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: Attack and defense strategies," *IEEE Networks*, vol. 20, pp. 41–47, 2006.
[38] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach," in *Proc. IEEE INFOCOM'04*, Mar. 2004.
[39] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," in *Proc. 10th ACM CCS'03*, Oct. 2003, pp. 62–72.
[40] CC2420 Datasheet, CC2420, 2.4 GHz IEEE 802.15.4/ZigBee-ready RF Tranceiver, Chipcon.
[41] Crossbow Technology. [Online]. Available: http://www.xbow.com
[42] The Sentilla Corporation. [Online]. Available: http://www.sentilla.com
[43] nesC. [Online]. Available: http://sourceforge.net/projects/nescc
[44] TinyOS. [Online]. Available: http://www.tinyos.net
[45] ZigBee Alliance. [Online]. Available: http://www.zigbee.org

**Sooyeon Shin** received the B.S. and M.S. degrees in computer engineering from Sejong University, Seoul, Korea, in 2004 and 2006, respectively. She is currently working towards the Ph.D. degree at Sejong University since 2006.

Her research interests include cryptography, computer network security, and wireless sensor network security.

**Taekyoung Kwon** (M'09) received the B.S., M.S., and Ph.D. degrees in computer science from Yonsei University, Seoul, Korea, in 1992, 1995, and 1999, respectively.

He was a Postdoctoral Research Fellow at the University of California, Berkeley, from 1999 to 2000. Since 2000, he has been contributing and working for the standardization in IEEE P1363.2 and ISO/IEC JTC1 SC27 11770-4. In 2001, he joined the Department of Computer Engineering, Sejong University, Seoul, Korea, where he is currently an Associate Professor of Computer Engineering. His research interests include information security, applied cryptography, cryptographic protocol, and wireless sensor network and its security issues.

**Gil-Yong Jo** received the B.S. and M.S. degrees in computer engineering from Sejong University, Seoul, Korea, in 2007 and 2009, respectively.

His research interests include cryptography, computer network security, and mobile network security.

**Youngman Park** received the B.S., M.S., and Ph.D. degrees in computer engineering from Hanyang University, Seoul, Korea in 1986, 1988 and 2004, respectively.

In 1990, he joined Korea Telecom (KT). He is now a Research Director at KT Central R&D Laboratory. His current research interest includes mobile cloud security, privacy preserving data mining and mobile TPM (Trusted Platform Module).

**Haegyu Rhy** (M'09) received the B.S. and M.S. degree in computer engineering from Seoul National University, Seoul, Korea, in 1989 and 1991, respectively. He is currently working towards the Ph.D. degree in computer engineering from Seoul National University, Seoul, Korea.

In 1992, he joined in Korea Telecom (KT) and is now working there. His research interests are authentication, ID management, security, trust, etc.