13

Lightweight Intrusion Detection for Wireless Sensor Networks

Eui-Nam Huh and Tran Hong Hai
Kyung Hee University
Republic of Korea

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 which 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, health-care, environment monitoring, etc. Some basic features of sensor networks are (Ilyas & Mahgoub, 2005):

– Self-organization
– Short-range broadcast communication and multi-hop routing
– 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 (Wood & Stankovic, 2002), Secure Routing in Sensor Networks (Karlof & Wagner, 2003). Current research on security in sensor networks generally focuses on secure routing protocols, key management and prevention techniques for specific attacks (Djenouri et al., 2005).
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 which protects the network from intrusion. In this chapter, 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 over 90% of malicious nodes under various attacks, with a high rate of packet collision. Our contribution 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.

Following this introduction section, the chapter is organized as follows: In the next section, 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 chapter ends with a conclusion and future work.

2. Security in wireless sensor networks

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 analysis on secure routing were performed by (Karlof & Wagner, 2003). 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 chapter we only examine inside attacks. Outside attacks are prevented by using link layer security mechanisms (Camtepe & Yener, 2005). They propose two types of adversaries, a mote-class adversary and laptop-class adversary. In the mote-class one, 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 one, 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 (Karlof & Wagner, 2003).

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 (Karlof & Wagner, 2003). 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.

Sinkhole: In a sinkhole attack, the adversary redirects nearly all the traffic from a particular area via a malicious node, creating a metaphorical sinkhole (Karlof & Wagner, 2003). 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 the data of member nodes in a cluster and relays them to another CH or the sink node.

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 (Khalil et al., 2005; 2008).

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.
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 (Camtepe & Yener, 2005; Sultana et al., 2007). 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

Intrusion Detection System (IDS) is defined as a system that tries to detect and alert of attempted intrusions into a system or a network (Richard Heady, 1990). 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 (Roesch, 2002; Paxson, 1999). The anomaly technique has the advantage that it doesn't require any rules and can detect novel attacks. The main disadvantage of anomaly detection is the high false positive rate (Balasubramaniyan et al., 1998; Cuppens & Miège, 2002; Janakiraman et al., 2003). 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 (Aboelaze & Aloul, 2005). A typical sensor node such as MICA has an 8 MHz microprocessor, 128 KB program flash memories and 512 KB serial flash memories (Technology, n.d.). 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. Though wireless Ad Hoc networks and wireless sensor networks share some common characteristics, and there was development of IDS in a wireless Ad Hoc network (Mishra et al., 2004), R. Roman showed in his paper that they can't be directly applied in WSNs (Roman, 2006). They proposed a novel technique for optimal monitoring of neighbors called spontaneous watchdog, which extends the watchdog monitoring mechanism in (Marti et al., 2000). 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. Ilker 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 (Onat & Miri, 2005). 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. Soumya et al. (2005) proposed an intrusion detection mechanism based on an ant colonies system (Banerjee et al., 2005). 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 P. et al published a paper in which they designed an intrusion framework based on the layout and selection of monitor nodes (Techateerawat & Jennings, 2006). They proposed a voting algorithm for selection of nodes which must trigger their IDS agent. Their approach reduced monitor nodes and energy consumption in networks, but also reduced the probability of detection. Unfortunately, their detection algorithms weren't demonstrated in detail. A recent study of Chong E. L. 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 (Chong Eik Loo & Palaniswami, 2006). A.P. Silva et al. proposed a decentralized IDS scheme, based on the specification in (da Silva et al., 2005). 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.

Afrand 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 (Agah et al., 2006). There are multiple IDS proposals for WSNs, but many are incomplete or only focus on a specific attack (Wang et al., 2006). Our contribution is based on previous works and involves the creation of a novel, efficient IDSs for WSNs. Furthermore, we propose a simple selection algorithm to trigger IDS modules in particular nodes. Our algorithm minimizes the monitor nodes which 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 (Abbasi & Younis, 2007)) 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 cluster heads are randomly selected and rotated to distribute the energy load among the network. In the steady phase, the cluster heads 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 which 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

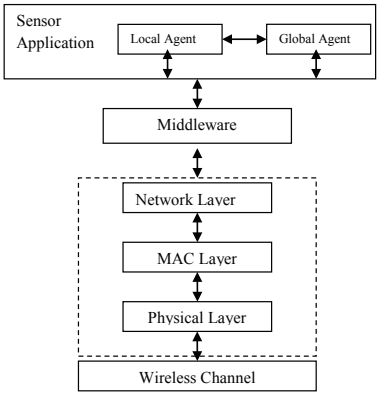


Fig. 1. Intrusion detection agents on sensor protocol's stack

participating nodes. Previous work on the application of IDS for sensor networks was undertaken by R. Roman (Roman, 2006). 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 (Khalil et al., 2005; da Silva et al., 2005; Hu & Burmester, 2009).

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.

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.

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 illustrated in Fig. 1.

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. 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 which contains information including the packet identification and type, source and destination, etc. Each

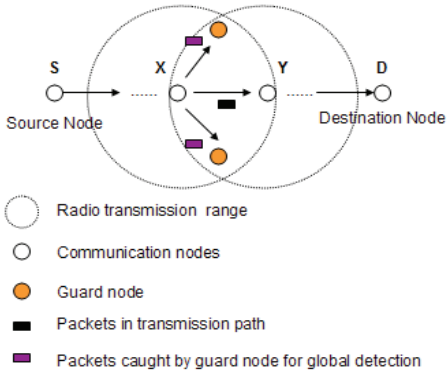


Fig. 2. Monitor node

entry in the buffer is time stamped. This expires after a timeout or after the entry in the buffer is examined by monitor nodes.

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

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 (Durresi et al., 2005). As we mentioned in Related Work, Issa Khail et al. applied two-hop neighbor knowledge to detect wormhole attacks in WSNs and Ad Hoc networks (Khalil et al., 2005; 2008). 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 neighbor, 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.

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

Source node	Intermediate node	Hop counter	.....

Fig. 3. Example of modified HELLO packet

```
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. 4. Algorithm of activating monitor nodes

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 (H. Debar & Feinstein, 2005).
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 previous work (da Silva et al., 2005). In our scheme, these rules are adapted to the routing protocols used.

- **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 which 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 Fig. 4. The global detection modules use two-hop neighbor knowledge and routing rules to detect anomalies within their transmission ranges. The illustration of Global_function() is represented in Fig. 5.
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 as follows:

```
Global_detection(packet_i)
1. {
2.  If Looking(packet_i_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. 5. Global detection at monitor nodes

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

Detection of Selective forwarding: In selective forwarding attacks, the transmission link from node A to node B is monitored by their monitor nodes, for example X, Y, Z. Node X, Y, 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 2 hops neighbor node's 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 forwarded to right path, the entry in the monitor node's intrusion buffer is remove.

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 base station by using high power transmission. All packets came to node A must be originated from A's neighbor list, the monitor nodes use neighbor's list and predefined signal rule to check if a packet is originated from a far located node.

Detection of Wormhole: Our system can detect four types of wormhole attacks by inherit the advantage of local monitoring mechanism. We use 2 hops neighbor's list and predefined rules to improve the detection of wormhole in clustered WSNs.

```
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. 6. Alert computation at the cluster-head

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. Tv_{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.

$$Tv_X = (Tv_{X,1}, Tv_{X,2}, ..., Tv_{X,N}) \tag{1}$$

Where $Tv_{X,i}$ represent the trust value of the i^{th} neighbor node of X. Calculation and update of reputation tables in sensor nodes can be found in (Kaplantzis et al., 2007). Our reputation system is fully adaptive with detection modules, because both schemes are based on an over-hearing mechanism. Each sensor node calculates the average trust of its neighbor nodes with the following equation:

$$E[X] = \frac{\sum_{i=1}^{N} Tv_{X,i}}{N} \tag{2}$$

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

$$Mp(Tv_{node}) = \left\{ \begin{array}{c} high - 0.8 \leq Tv_{node} \leq 1 \\ medium - 0.5 \leq Tv_{node} \leq 0.8 \\ uncertain - 0.3 \leq Tv_{node} \leq 0.5 \\ low - 0 \leq Tv_{node} \leq 0.3 \end{array} \right\} \tag{3}$$

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.

```
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. 7. Improved alert computation algorithm at the CHs

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 which 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 malicious node is described as follows:

$$MC_{node} = \beta \sum_{j=1}^{i} i + \delta \sum_{k=1}^{k} j + \varphi \sum_{l=1}^{l} k \tag{4}$$

Where $0 < \beta < \delta < \varphi < 1$ and i, j, k are the number of alert packets with the correlative trust states mentioned above. So, aggregation and computation of alert packets at CHs is improved as Fig. 7 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 Selection algorithm

As mentioned in the previous section, the monitor nodes observe the behavior's packet that pass through them to destination. To minimize the number of nodes activating the intrusion detection modules, our proposed scheme select the nodes which cover as many other nodes as possible. Our main idea is to choose the set of nodes which corporately cover all the nodes in the networks. Our proposed scheme is based only on the neighbor node information built on each node to find these nodes. We also make the assumption that the adversary cannot successfully compromise a node during the short deployment phase. Thus, the neighbor node information sent to sink node is trustful. The selection of monitor nodes is performed by sink node by following process:

At sink node

```
Assign U ={R} / List of nodes in
      networks

Repeat

  For each node i in U

  Find Max N(i) in U

  Put i in stack

  Assign U = U | N(i)

Until U = Null

Send IDS request to nodes in stack
```

Fig. 8. Selection algorithm at sink node

- After deployment, the sensor node builds its direct neighbor node's list and sends it to the sink node.
- The sink node finds the set of nodes which corporately cover all nodes in the network as the chosen monitor nodes. The finding algorithm is explained in detail below.
- The sink node sends the request message to these chosen nodes to require them activating their intrusion detection modules.
- Every message sent by sensor node or sink node is authenticated by using their shared keys.

We consider a network of N sensors as a set of static nodes denotes as and a single sink node denoted as $R = \{n_1, n_2, ..., n_N\}$. To describe selection algorithm, we use the term "sensor" and "node" interchangeably. The communication in the network is always destined toward the sink node . Nodes i and j are neighbors if they are in its radio range, denoted by an edge $(i\ j)$. Let $N(i) := j|(i,j)$ denote the set of neighbors of node i and $N(i)|j$ denote the set without node j. Besides, we assume sink node or cluster heads (CHs) can have a greater battery powers, a more capable CPU or a sensitive antenna which can reach to other CHs or the sink node. The sink node search for the set of nodes which corporately cover all nodes in the network based on their neighbor node information received. The algorithm is described in Fig. 8.

4.4 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. Fig. 9 illustrates the case when two monitor nodes X, Z send the same alert packet about a malicious node Y.

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 (Bianchi, 2000), where nodes use over-hearing to determine when the channel is free. In (Le et al., 2006), 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 doesn't 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 (Hill et al., 2000a;b) found that each bit transmitted in WSNs consumes power about equivalent to executing 800-1,000 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 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 the 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 which has the probability of more than X nodes sending an alert. Because the events are independent so

$$P_D = P_X + P_{X+1} + ... + P_K \tag{5}$$

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

$$P_X = (1 - P_C)^X {P_C}^{K-X} \tag{6}$$

So the probability detection of an attacker PD can be written as following:

$$P_D = (1 - P_C)^X {P_C}^{K-X} + ... + (1 - P_C)^K {P_C}^{K-K} \tag{7}$$

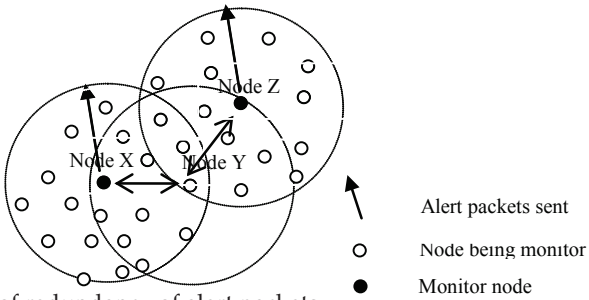


Fig. 9. Illustration of redundancy of alert packets

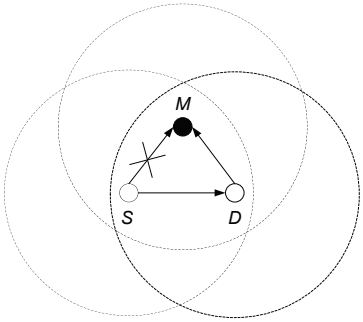


Fig. 10. False positive detection

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

$$P_D = \sum_{i=X}^{K} \begin{pmatrix} K \\ X \end{pmatrix} (1 - P_C)^X {P_C}^{K-X} \tag{8}$$

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 doesn't 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 Fig. 10 illustrates the false positive of a monitor node. The probability of false detection of monitor node M can be found as following steps:
$P_F = P_S + P_D$, 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 following:

$$P_S = {P_C}^2(1 - P_C) \tag{9}$$

The probability of P_D can be written as following:

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

$$\Rightarrow P_F = (1 - P_C)^2 P_C + {P_C}^2(1 - P_C) \tag{11}$$

Similar to equation (8), we have the false probability of monitor nodes:

$$\Leftrightarrow P_{FD} = \sum_{i=X}^{K} \begin{pmatrix} K \\ X \end{pmatrix} (1 - P_F)^X {P_F}^{K-X} \tag{12}$$

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

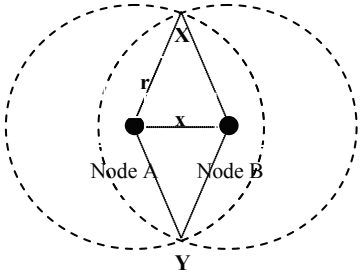


Fig. 11. The radio coverage of two communication nodes

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 (the Fig. 11).
For any distance x, the radio coverage of two communication nodes is the area of the sectors XAY and XBY minus the area of the rhombus AXBY and is calculated as following:

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

The probability distribution function of x is given by

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

So the probability density function is

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

The expected area XY is calculated as following:

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

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

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

So the average number of monitor nodes for each individual link is given by $[E[XY] \times d$, where d is network density. As shown in Fig. 12, 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 Fig. 13, 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.

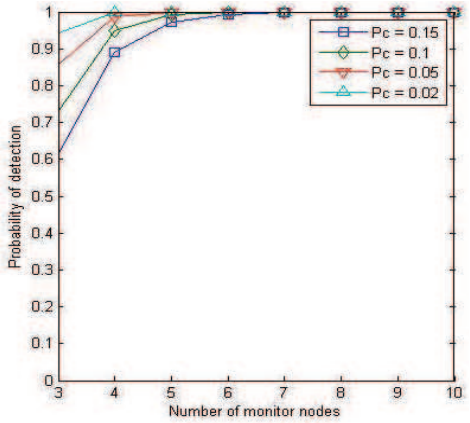


Fig. 12. Detection probability of a malicious node

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 meters x 100 meters, using Castalia, a WSNs simulator based on Omnet++ (Castalia Simulator). The parameters used are in accordance with actual sensor network applications and experiments, such as Smart Dust Project (2001), Virtual Patrol (2005) (Gui & Mohapatra, 2005). 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 packets, sensor nodes must call the detection algorithm before forwarding or receiving the data. To simplify algorithms, we assign each sensor node a random trust value. There is no low-trust value during the periods of deployment.

Fig. 14 shows the performance of our scheme with malicious nodes. Castalia also supports packet collision by setting the parameter *SN.WirelessChannel.CollisionModel* (Castalia Simulator). We set sensor nodes to exhibit malicious behavior by increasing their dropped packet ratio, changing the fields of forwarded packets and sending false Hello

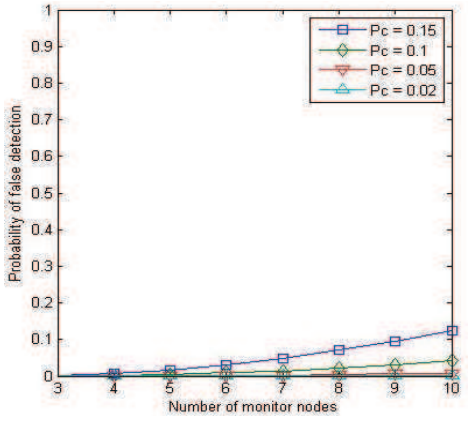


Fig. 13. False detection probability of a malicious node

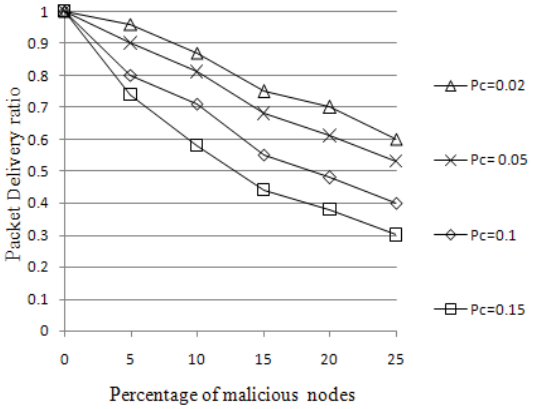


Fig. 14. Packet delivery ratio under attacks

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 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 Fig. 15, 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.

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 (Khalil et al., 2005; 2008; Roman,

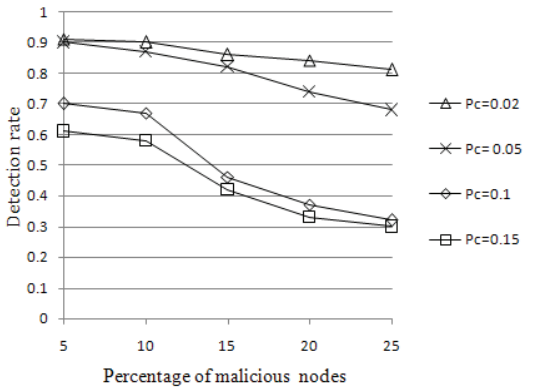


Fig. 15. Detection ratio of malicious nodes

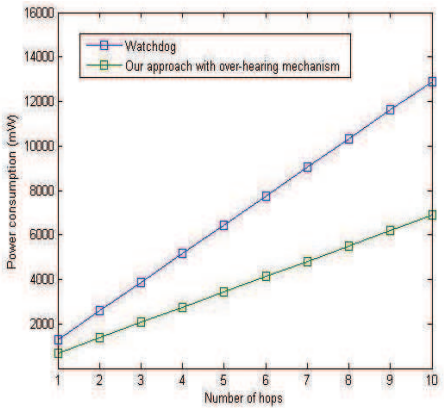


Fig. 16. Power consumption comparison

2006; Chong Eik Loo & Palaniswami, 2006; Hu & Burmester, 2009; Marti et al., 2000; Kaplantzis et al., 2007; Hai et al., 2007). 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 (Hai et al., 2007; Holger & W, 2005), we obtain the energy consumption of monitor nodes in the transmission link in Fig. 16 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 1 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 Afrand's work (Agah et al., 2006), a detection framework was proposed, based on

IDS framework / **Characteristic**	Our proposed scheme	Onat's scheme	Chong's scheme	Afrand's scheme
Architecture	Distributed & Collaboration	Distributed	Distributed	Distributed
Approach	Major voting, two-hop neighbor knowledge, routing rules	Traffic model & Centralized detection	Traffic model & 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

Table 1. A review of related works on intrusion detection

non-cooperative games, but the detection algorithms were not shown in detail.

6. Conclusion

In this chapter, 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.

7. References

Abbasi, A. A. & Younis, M. (2007). A survey on clustering algorithms for wireless sensor networks, *Comput. Commun.* 30(14-15): 2826–2841.

Aboelaze, M. & Aloul, F. (2005). Current and future trends in sensor networks: a survey, *Wireless and Optical Communications Networks, 2005. WOCN 2005. Second IFIP International Conference on*, pp. 551 – 555.

Agah, A., Basu, K. & Das, S. K. (2006). Security enforcement in wireless sensor networks: A framework based on non-cooperative games, *Pervasive Mob. Comput.* 2(2): 137–158.

Balasubramaniyan, J. S., Garcia-Fernandez, J. O., Isacoff, D., Spafford, E. & Zamboni, D. (1998). An architecture for intrusion detection using autonomous agents, *ACSAC '98: Proceedings of the 14th Annual Computer Security Applications Conference*, IEEE Computer Society, Washington, DC, USA, p. 13.

Banerjee, S., Grosan, C. & Abraham, A. (2005). Ideas: Intrusion detection based on emotional ants for sensors, *In 5th International Conference on Intelligent Systems, Design and Applications (ISDA-05*.

Bianchi, G. (2000). Performance analysis of the ieee 802.11 distributed coordination function, *Selected Areas in Communications, IEEE Journal on* 18(3): 535 –547.

Camtepe, S. A. & Yener, B. (2005). Key distribution mechanisms for wireless sensor networks: a survey, *Technical report*.

Castalia Simulator *http://castalia.npc.nicta.com.au*.

Chong Eik Loo, Mun Yong Ng, C. L. & Palaniswami, M. (2006). Intrusion detection for routing attacks in sensor networks, *International Journal of Distributed Sensor Networks*, Vol. 2, pp. 313 – 332 Vol. 3.

Cuppens, F. & Miège, A. (2002). Alert correlation in a cooperative intrusion detection framework, *SP '02: Proceedings of the 2002 IEEE Symposium on Security and Privacy*, IEEE Computer Society, Washington, DC, USA, p. 202.

da Silva, A. P. R., Martins, M. H. T., Rocha, B. P. S., Loureiro, A. A. F., Ruiz, L. B. & Wong, H. C. (2005). Decentralized intrusion detection in wireless sensor networks, *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, ACM, New York, NY, USA, pp. 16–23.

Djenouri, D., Khelladi, L. & Badache, A. (2005). A survey of security issues in mobile ad hoc and sensor networks, *Communications Surveys Tutorials, IEEE* 7(4): 2 – 28.

Durresi, A., Member, S., Paruchuri, V. K., Member, S., Iyengar, S. S. & Kannan, R. (2005).

Optimized broadcast protocol for sensor networks, *IEEE Transactions on Computers* 54: 1013–1024.

Gui, C. & Mohapatra, P. (2005). Virtual patrol: a new power conservation design for surveillance using sensor networks, *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pp. 246 – 253.

H. Debar, D. C. & Feinstein, B. (2005). The intrusion detection message exchange format.

Hai, T. H., Khan, F. & Huh, E.-N. (2007). Hybrid intrusion detection system for wireless sensor networks, *ICCSA'07: Proceedings of the 2007 international conference on Computational science and Its applications*, Springer-Verlag, Berlin, Heidelberg, pp. 383–396.

Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D. & Pister, K. (2000a). System architecture directions for networked sensors, *SIGPLAN Not.* 35(11): 93–104.

Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D. & Pister, K. (2000b). System architecture directions for networked sensors, *SIGARCH Comput. Archit. News* 28(5): 93–104.

Holger, K. & W, W. A. (2005). *Protocols and Architecture for Wireless Sensor Networks - Chapter 2.2*, John Wiley & Son Press.

Hu, J. & Burmester, M. (2009). Cooperation in mobile ad hoc networks, *Guide to Wireless Ad Hoc Networks*, Computer Communications and Networks, Springer London, pp. 1–15. 10.1007/978-1-84800-328-6_3.
http://dx.doi.org/10.1007/978-1-84800-328-6_3

Ilyas, M. & Mahgoub, I. (2005). *Handbook of sensor networks: Compact wireless and wired sensing systems*, CRC Press.

Janakiraman, R., Waldvogel, M. & Zhang, Q. (2003). Indra: a peer-to-peer approach to network intrusion detection and prevention, *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on*, pp. 226 – 231.

Kaplantzis, S., Shilton, A., Mani, N. & Sekercioglu, Y. A. (2007). Detecting selective forwarding attacks in wireless sensor networks using support vector machines.

Karlof, C. & Wagner, D. (2003). Secure routing in wireless sensor networks: attacks and countermeasures, *Ad Hoc Networks* 1(2-3): 293 – 315. Sensor Network Protocols and Applications.
http://www.sciencedirect.com /science /article /B7576-499CSFN-7 /2 /ad3f92c2d573d82839cdb5ae91272fd7

Khalil, I., Bagchi, S. & Shroff, N. (2005). Liteworp: a lightweight countermeasure for the wormhole attack in multihop wireless networks, *Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on*, pp. 612 – 621.

Khalil, I., Bagchi, S. & Shroff, N. B. (2008). Mobiworp: Mitigation of the wormhole attack in mobile multihop wireless networks, *Ad Hoc Netw.* 6(3): 344–362.

Le, H.-C., Guyennet, H. & Zerhouni, N. (2006). Over-hearing for energy efficient in event-driven wireless sensor network, *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, pp. 633 –638.

Marti, S., Giuli, T. J., Lai, K. & Baker, M. (2000). Mitigating routing misbehavior in mobile ad hoc networks, *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, ACM, New York, NY, USA, pp. 255–265.

Mishra, A., Nadkarni, K. & Patcha, A. (2004). Intrusion detection in wireless ad hoc networks, *Wireless Communications, IEEE* 11(1): 48 – 60.

Onat, I. & Miri, A. (2005). An intrusion detection system for wireless sensor networks, *Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE*

International Conference on, Vol. 3, pp. 253 – 259 Vol. 3.

Paxson, V. (1999). Bro: A system for detecting network intruders in real-time, *Computer Networks*, pp. 2435–2463.

Richard Heady, George Lugar, M. S. A. M. (1990). The architecture of a network level intrusion detection system, *Technical report*, University of New Mexico, Albuquerque, NM.

Roesch, M. (2002). The snort network intrusion detection system. *http://www.snort.org*

Roman, R. (2006). Applying intrusion detection systems to wireless sensor networks, *in CCNC 2006: Proceeding of the 3rd IEEE Consumer Communications and Networking Conference*, pp. 640–644.

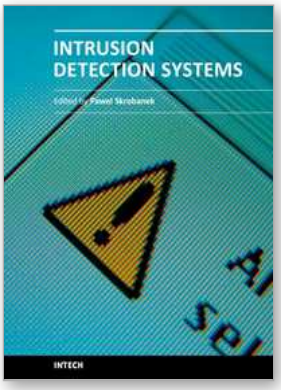
Sultana, N., Choi, K.-M. & Huh, E.-N. (2007). Application driven cluster based group key management with identifier in mobile wireless sensor network, *FGCN '07: Proceedings of the Future Generation Communication and Networking*, IEEE Computer Society, Washington, DC, USA, pp. 362–367.

Techateerawat, P. & Jennings, A. (2006). Energy efficiency of intrusion detection systems in wireless sensor networks, *Web Intelligence and Intelligent Agent Technology Workshops, 2006. WI-IAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on*, pp. 227 –230.

Technology, C. (n.d.). Mica2, wireless measurement system. *http://www.xbow.com*

Wang, Y., Attebury, G. & Ramamurthy, B. (2006). A survey of security issues in wireless sensor networks, *Communications Surveys Tutorials, IEEE* 8(2): 2 –23.

Wood, A. D. & Stankovic, J. A. (2002). Denial of service in sensor networks, *Computer* 35(10): 54–62.



Intrusion Detection Systems

Edited by Dr. Pawel Skrobanek

ISBN 978-953-307-167-1
Hard cover, 324 pages
Publisher InTech
Published online 22, March, 2011
Published in print edition March, 2011

The current structure of the chapters reflects the key aspects discussed in the papers but the papers themselves contain more additional interesting information: examples of a practical application and results obtained for existing networks as well as results of experiments confirming efficacy of a synergistic analysis of anomaly detection and signature detection, and application of interesting solutions, such as an analysis of the anomalies of user behaviors and many others.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821