

A Multi-Layer Intruder Detection System for Multi-Hop Cluster-Based Sensor Networks

Ameer Ahmed Abbasi
Al-Hussan Institute of Management &
Computer Science
Dammam-31411 Saudi Arabia
ameer_abbasi@hussan.edu.sa

M. I. Buhari
Department of Information & Computer Science
King Fahd University of Petroleum & Minerals
Dhahran-31261 Saudi Arabia
mibuhari@ccse.kfupm.edu.sa

M. Akbar Badhusha
Faculty of Information Technology
University of Technology
Sydney, Australia
akbar@it.uts.edu.au

Abstract

Wireless Sensor Networks (WSNs) are playing a fundamental role in emerging pervasive platforms that have potential to host a wide range of next generation civil and military applications. Inexpensive sensor nodes are deployed to the sensing area with little mobility and high density. Furthermore, in many scenarios WSNs are of interest to adversaries and they become susceptible to some types of attacks since they are deployed in open and unprotected environments and are constituted of cheap small devices. Since preventive mechanisms provide no surety to hold the intruder, an intrusion detection system (IDS) running beside preventive mechanism will greatly increase the security of network. Introducing IDS in wireless media is not as similar to that of wired systems due to the differences in capturing audit data and availability of resources. This paper proposes a method of implementing the IDS on cluster-based WSNs using multi-hop data communication.

Keywords

Multi-Hop cluster-based WSNs, Security, Intrusion detection, Gateway nodes, Sensor nodes

1. Introduction

Wireless Sensor Networks (WSNs) are widely used in military surveillance, building security, and harsh physical environments. Inexpensive sensor nodes are deployed to the sensing area with little mobility and high density. The sensor nodes have very limited battery power and computing capability. Hence, the sensor networks should be well organized to meet the task. Clustering of sensor nodes into groups with proper clusterhead (gateway) selection will impose a regular structure in the sensor network. It is easier to control a gateway with the summarized information instead of handling information from every sensor node individually. Thus, gateways aggregate the data from various nodes and communicate with the far distant base station. Nodes usually share the same physical media; they transmit and acquire signals at the same frequency band, and follow the same hopping sequence or spreading code. The data-link-layer functions manage the wireless link resources and coordinate medium access between the node and its clusterhead. The medium access control (MAC) protocol is essential to cluster-based WSNs because it allows nodes to share a common broadcast channel.

The nature of cluster-based WSNs makes them very vulnerable to an adversary's malicious attacks. Since cluster-based WSNs are based on wireless communication links, an attack can come from all directions and target any node. These attacks such as denial of service, message negligence, exhaustion, black-hole, HELLO flood etc. could damage network seriously. Most routing protocols in cluster-based WSNs and ad-hoc networks are cooperative in nature and presents vulnerability [1]. An adversary who hijacks a sensor node could paralyze the entire wireless network by disseminating false routing information. Intrusion prevention measures, such as encryption and authentication, can be used in cluster-based WSNs to reduce intrusions, but cannot eliminate them. For example,

encryption and authentication cannot defend against compromised nodes, which carry the private keys. Integrity validation using redundant information (from different nodes), such as those being used in secure routing [2, 3], also relies on the trustworthiness of other nodes, which could likewise be a weak link for sophisticated attacks.

Some common types of attacks to sensor networks are presented in [5, 8]. Many researchers have presented preventive mechanisms that can be applied to protect sensor networks against some types of attacks [4, 7]. However, there are some attacks for which there are no known prevention methods, such as wormhole [5, 6]. Since preventive mechanisms provide no surety to hold the intruder, an intrusion detection system (IDS) running beside preventive mechanism will greatly increase the security of network. Development of IDS for cluster-based WSNs poses many challenges to the researcher, mainly due to severe restricted resources. The methods developed for traditional networks demand resources and cannot be used as is in cluster-based WSNs. Normally, most of the cluster-based WSNs are application oriented and have very specific characteristics according to the target application. An anomaly or unexpected behavior of the cluster-based WSNs indicates an attack and most of the IDS use these anomalies to detect an intrusion in the target system.

In this paper, we propose a new model of IDS for cluster-based WSNs using multi-hop data communication. Our main contributions pertain towards: the proposal of a multi-layer IDS model for cluster-based WSN restrictions and peculiarities; a high-level methodology to construct specific IDS to a target cluster-based WSN with well defined applications. The proposed IDS is based on the inference of the network behavior obtained from the analysis of events detected by monitor nodes. Data messages listened to by the monitor node but not addressed to it along with message collisions that occur when the monitor node tries to send messages, are considered. In summary, to prevent inherent vulnerabilities of any cluster-based WSN, we need to deploy IDS techniques in addition to preventive mechanisms. Further research is necessary to adapt these techniques to this new environment from their original applications in fixed wired network.

The rest of the paper is organized as follow: Section 2 briefly describes the related work. Section 3 presents the rules proposed for our IDS. Section 4 discusses the system architecture model and some design issues associated with the IDS. Section 5 discusses the proposed algorithm used in the IDS. Section 6 discusses the concerns of our proposed algorithm. Finally, Section 7 presents the conclusion for this work.

2. Related Work

Intrusion detection is an important issue in the area of networks security. There are vast differences between the traditional fixed networks and cluster-based WSNs. It is very difficult to apply intrusion detection techniques proposed and developed for a traditional fixed network [9, 10, 11, 12, 13, 14] to a cluster-based WSN. Having resemblance with cluster-based WSNs, ad-hoc networks also suffer in severe resource restrictions although they are not as restrictive as cluster-based WSNs. There are many proposed solutions available to intrusion detection in ad-hoc networks, but too little has been contributed in relation to cluster-based WSNs. In [6], a method is proposed for detecting wormhole attacks in ad-hoc networks. The method works by evaluating the time spent on the transmission of packets between nodes in the network, and by node authentication. The paper proposes two protocols: *Slot Authenticate MAC* and *TIK* that require network trusted time synchronization, it is not appropriate to consider it for our work. Keeping nodes synchronized in a cluster-based WSN is very hard even if it is possible. In [15], a behavioral paradigm and a decentralized cluster-based IDS model for ad-hoc networks are presented. A technique is presented and explained to elect a responsible node to monitor intrusion detection cycles. In context of cluster-based WSNs, the solution is expensive and inadequate. In [16], another method is proposed to detect attacks in WSNs. The method compares the power of received signal with the power of observed signal in the network to detect attacks like HELLO flood, wormhole etc. Our aim is to propose a wider solution, capable of detecting several types of intruder and attacks. The strategy proposed in [16] still can be used as one of the rules of our propose system. In [17], the idea of a watchdog is introduced to improve the intrusion detection in ad-hoc networks. This idea inspired us to introduced IDS at each cluster separately to watch each and every node in it. If any node in the cluster changes, delays, replicates, or simply keeps the message that should be retransmitted, the gateway for the cluster counts it as a failure. This technique would also detect other types of attacks. In [18], a fault-tolerant solution based on route redundancy is presented that keeps the network functioning even on the presence of intruders. However, many of the attacks found on literature cannot be tolerated. This fact motivates the development of IDS adequate to cluster-based WSNs.

3. Rules and Definitions

Development of appropriate IDS to a target cluster-based WSN can be divided into three following important steps: (1) pre-select, from the available set of rules, those that can be used to monitor the features defined by the designer;

(2) compare the information required by the pre-selected rules with the information available at the target network to select rules definitively; and (3) set the parameters of the selected rules with the values of the design definitions. Definitions of the available rules are presented in the following:

Integrity Rule: to avoid data fusion or aggregation by other sensor nodes, the message payload must be the same along the path from its origin to a destination. Attacks where the intruder modifies the contents of a received message can be detected by this rule.

Jamming Rule: the number of collisions associated with a message must be lower than the expected number in the network. The jamming attack, where a node introduces noise into the network to disturb the communication channel, can be detected by this rule.

Interval Rule: if the time interval between the receptions of two consecutive messages is longer or shorter than the allowed time limits, a failure is raised. Two attacks that will probably be detected by this rule are the negligence attack and the exhaustion attack. In the negligence attack, the intruder does not send data messages generated by a tampered node. While in the exhaustion attack, the intruder increments the message-sending rate in order to increase the energy consumption of other nodes in the cluster.

Repetition Rule: the same message can be retransmitted by a node only a limited number of times. This rule can detect an attack where the intruder sends the same message several times, thus promoting a denial of service attack.

Radio Transmission Range: all messages listened to by the monitor node must be originated from one of the nodes within its cluster. Attacks like wormhole and hello flood, where the intruder sends messages to a far located node using a more powerful radio, can be detected by this rule.

Retransmission Rule: the monitor listens to a message, pertaining to one of its neighbors as its next hop, and expects that this node will forward the received message, which does not happen. Two types of attacks that can be detected by this rule are the blackhole and the selective forwarding attack. In both of them, the intruder suppresses some or all messages that were supposed to be retransmitted, preventing them from reaching their final destination in the network.

Delay Rule: the retransmission of a message by a monitor's neighbor must occur before a defined timeout. Otherwise, an attack will be detected.

In the retransmission, integrity, delay, repetition and interval rules, the monitor node suspects about the nodes in its cluster. Using this technique, besides detecting an attack, we can find out both the address and location of the intruder in any cluster. Each rule discussed here is given weights. The rule that is critical is given higher weight compared to other rules. Make sure that the minimum most weight assigned is 1.

4. System Architecture Model

The system architecture for the sensor network is depicted in Figure 1. In the architecture, by default, sensor nodes are grouped into clusters controlled by a base station. Every cluster has a gateway node that manages sensors in the cluster. Clusters can be formed based on many criteria such as communication range, number and type of sensors and geographical location [19, 20]. In our model, the gateways collaboratively locate the deployed sensors and group them into clusters so that sensors' transmission energy is minimized while balancing the load among the gateways [21, 22, 23]. In this paper, we assume that sensor and gateway nodes are stationary. Sensor nodes in a cluster can send data packets to the gateway node directly if they are very close to it. In case the gateway is not very near, sensor nodes can use multi-hop data communication to minimize transmission energy. Since the system architecture supports multi-hop data communication, it is assumed that the sensor can act as

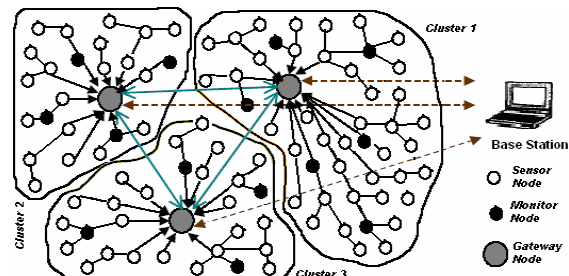


Figure 1: The architecture for cluster-based WSN using multi-hop data communication.

a relay to forward data from another sensor. We also assume that the base station (which could be located in a helicopter or van or anywhere else) can locate all the deployed gateways and group them into a “virtual cluster” as illustrated in figure 2. Virtual cluster exists logically and is managed, controlled and monitored by the base station. In our architecture, all the gateway nodes in the cluster-based WSN are members of the virtual cluster. The base station receives local intrusion detection reports from gateway nodes and is responsible to monitor and detect system-wide intrusions. It also performs the gateway function of the virtual cluster, in addition to its responsibilities as a base station.

We have considered five types of nodes in the system: sensor nodes, monitor nodes, intruder nodes, gateway nodes and the base station. The *sensor nodes* have sensor and router functions. As a sensor, it collects sensing data, and sends it to the gateway node. As a router, it retransmits all messages directed to the gateway node. The *monitor node* is responsible for monitoring its neighbors looking for intruders. By doing this, the node keeps its radio in a promiscuous mode, storing relevant information and, after a given time period, sends it to gateway node for further processing. This node also executes the sensor/router functions since it is a common sensor node where an IDS data collector unit is installed. The *gateway nodes*, which are significantly less energy-constrained than the sensors, interface the base station with the sensor network via long-haul communication links. The gateway node is also responsible to select the monitor nodes randomly from its cluster to perform the IDS data collector unit function. It receives information from these IDS data collector units and further processes this information according to selected rules. The gateway node sends intrusion detection reports to the base station in addition to the reports generated through fusion of sensor readings, e.g. tracks of detected targets. The *intruder node* switches between a sensor node behavior and an intruder behavior. The intruder behavior depends on the considered attack. The intruder can spend from 1% to 100% of its time performing an attack. The *base station* performs system-level fusion of collected reports for overall situation awareness and also acts as gateway of virtual cluster (a logical cluster of gateway nodes).

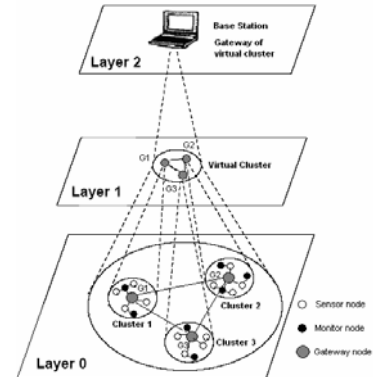


Figure 2: Model of IDS with virtual-cluster

Our proposed IDS model (Figure 2) is distributed and cooperative in nature. Each gateway node is responsible for detecting signs of intrusion, by processing information received from IDS data collector units, in its cluster locally and independently. The IDS agent at gateway node is responsible to analyze local activities (including sensing and systems activities, and communication activities) within the cluster. It detects intrusion from local traces and initiates response. The base station is given due rights to monitor and detect signs of intrusion in the virtual cluster. The IDS agent at base station runs autonomously, receives intrusion detection reports from gateway nodes and monitors activities (including node data collection, intrusion detention reports, processing and forwarding activities, system and communication activities) within the virtual cluster. The base station involvement with the virtual cluster provides another layer of IDS above from the gateway nodes. This upper layer provides system-wide intrusion detection and protects the network from a gateway node that starts behaving as an intruder itself. In the systems aspect, the base station and each and every gateway node of the cluster-based WSN run an IDS agent. Note that since the base station and all the gateway nodes are required to act as IDS agents too, it is necessary to have the trained IDS agents pre-installed on all of them. Since a gateway node can select any node in its cluster as to perform monitor node activities, all nodes in a sensor network should be equipped with an IDS data collector unit.

5. Proposed Algorithm Model

Our proposed algorithm model for intrusion detection is divided into the following three units: (i) Data collection unit (ii) Data analysis unit and (iii) Intrusion detection engine. During each round of intrusion detection, every IDS agent goes through these three units before making a decision to raise intrusion detection alert or not. Figure 3 shows the architecture of a gateway node. As we mentioned earlier, the IDS agent at a gateway node is responsible to receive captured messages from the IDS data collector units at monitor nodes and further processes these messages according to selected rules. This node runs the gateway node functions, like interfacing the base station with the sensor network via long-haul communication links, sending reports to the base station which are

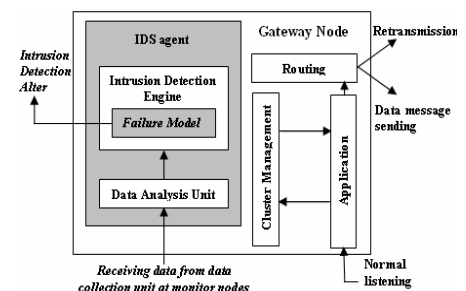


Figure 3: Gateway node with IDS

generated through fusion of sensor readings etc., in addition to the IDS agent functions. The base station, which is also acting as a gateway to the virtual cluster, performs the base station tasks in addition to IDS agent functions.

5.1 Data Collection Unit

The data collection unit of an IDS agent runs at monitor nodes and listens to the messages from its cluster promiscuously, filters important information based on the rules discussed in section 3 and stores the filtered information in an array data structure for subsequent analysis. Here important information means the message fields that might be useful to the rules application unit. Messages to which applying rules cannot be applied are not stored. Using this technique each data collection unit saves memory, processing time and energy of its host platform. Data extracted from the messages are sent to gateway node for further processing according to selected rules either after a given period of time or when there is no space left in memory. From the monitor's point of view, the time is divided in slices. Each time slice starts when the array is empty and begins to store messages captured in promiscuous mode. The time slice finishes when the array is completely full and the stored messages can be sent to gateway node for further processing. The length of the array defines the size of the time slice when the monitor is hearing in promiscuous mode, and, thus, defines the amount of messages that can be related to each other in order to find an intruder. There is a trade off between the storage cost and the detection efficiency. If the array is small and, consequently, its storage cost, the time slices will be small and the message sequence break will be large, which implies worse detection efficiency.

5.2 Data Analysis Unit

This unit runs at gateway node and the rules are sequenced in descending order based on their weights. This sequence of rules is applied to each data entry in the array data structure received from data collector units. These rules are specific to each message type. If a message fails in one of the rules, no other rules will be applied to it and message can be discarded. In addition, the failure counter of that node is incremented based on the weight to use it in intrusion detection stage. This strategy makes sense because the first failure already gives an indication of an abnormal behavior in the network and it is not appropriate to evaluate the message further as it is well known that the cluster-based WSNs have severe resource restrictions. This strategy also reduces the detection latency with a trade off between the accuracy, processing cost, and running time. After being tested against all rules without failing any of them, the message is discarded. Algorithm 1 summarizes the rule application technique described above.

Algorithm 1: Rules application unit of IDS agent

```
1: for all messages in data structure array do
2:     for all rules specific to the message in descending order by weight do
3:         apply rule to the message;
4:         if (message == fail) then
5:             increment failure counter for the node based on weight;
6:             [failure counter = failure counter + weight]
7:             discard message;
8:             break;
9:         end if
10:    end for
11: end for
```

5.3 Intrusion Detection Engine

In order to implement IDS which is capable of, in most cases, distinguishing occasional network failures from attack instances promoted by intruders, we have proposed a solution in which a gateway node can infer the purpose of a suspect node participating on the network cluster, since these failures are similar to the attacks. The following solution addresses the issues raised by attacks such as denial of service, message negligence, exhaustion, black-hole, HELLO flood, and jamming. In our model, an attack is raised only if, after counting all the cluster failures detected by the gateway node during the analysis of messages transmitted on its cluster in a round, its number is greater than

an expected value. The gateway node calculates this expected number dynamically, using a failure history for each node in its cluster. An average of the number of failures that have occurred since the deployment of the sensor nodes is kept and updated every time the IDS are activated. The history update takes place only if the number of failures for that round is close to the cumulative value kept by the gateway node. In this case, the value of the round failure and the previous cumulative value are combined to form a new cumulative value. This technique introduces the idea of a deviation tolerance. Although occasional failures may happen during each round of message capture by the gateway nodes, its number is not known beforehand. By determining the variance bounds for it, an IDS can raise an attack indication whenever these limits are reached. In other words, an attack indication is only signaled by the gateway node when an abnormal behavior occurs with a frequency higher than expected. Considering that the failure expectancy takes time to stabilize, a large number of false positives will appear at the beginning of the network life cycle. To avoid this, a learning period (which includes only the data collection unit) at the beginning of the network life cycle will be introduced. In this period, a gateway node will not consider any abnormal event, for a certain time, in order to prevent an attack indication from being mistakenly signaled, while the average has not settled down. This learning period must not last long, or else the damage promoted by possible intruders on the network can be overwhelming. Algorithm 2 summarizes the technique described above.

Algorithm 2: Intrusion detection engine comparing round-failure average with failure history

```
1: for all cluster nodes do
2:   for all failure types do
3:     if round-failure value > cumulative value then
4:       signal attack indication
5:     else
6:       update cumulative value by combining it with round-failure value
       [current failure value is given more weight than old history value]
       [cumulative value = cumulative value * 0.1 + round-failure value * 0.9]
7:     end if
8:   end for
9: end for
```

6. Concerns of our Proposed Algorithm

In our proposed distributed IDS model, every gateway node in the cluster-based WSN participates in intrusion detection. Each gateway node is responsible for detecting signs of intrusion in its cluster locally and independently. Another possibility is that the gateway node may itself compromise. To overcome this problem, the base station is given due rights to monitor and detect signs of intrusion in the virtual cluster. The base station involvement with the virtual cluster provides another layer of IDS above the gateway nodes. Furthermore, we have considered some rules and definitions specific to the cluster-based WSNs, using which a gateway node suspects about the nodes in its cluster. These rules and definitions provide a solution that addresses the issues raised by attacks such as denial of service, message negligence, exhaustion, black-hole, HELLO flood, and jamming. A new type of attack would be able to trace pass the proposed IDS possibly, if no update was performed on the rules and definitions.

In this algorithm, we have proposed the idea of using weights to consider the critical issues seriously. Each rule discussed here is given weights. The rule that is critical is given higher weight compared to other rules. Make sure that the minimum most weight assigned is 1. Also, we plan to give more weight for recent happenings those historical values. Thus, we have updated the cumulative values as shown in Algorithm 2. The above-mentioned are proposed concepts and thus needs to be validated using simulation.

7. Conclusion

In this paper, we have discussed about handling intrusion detection in cluster-based WSNs that support multi-hop data communication. In order to apply intrusion detection in an efficient way, we have proposed the implementation of IDS data collector unit at monitor nodes and IDS agents at gateways. Formation of virtual cluster among gateway nodes is also proposed. The base station is given due rights to make sure that the gateways themselves do not become intruders. Applying well-known intrusion detection techniques to cluster-based WSNs raises concerns and forces us to generate the kind of intrusion that needs less traffic to learn and fewer rules to apply. By considering weights for different rules, we respect the influence of each rule on our system.

8. References

- [1] Royer, E. and Toh, C.-K. (1999), A review of current routing protocols for ad-hoc mobile wireless networks, *IEEE Personal Communication*, 6(2), April 1999, pp. 46 – 55.
- [2] Smith, B. R., Murthy, S., and Garcia-Luna-Aceves, J. (1997), Securing distance-vector routing protocols, In *Proceedings of Internet Society Symposium on Network and Distributed System Security*, San Diego, California, February 1997, pp. 85 – 92.
- [3] Zhou, L. and Haas, Z. J. (1999), Securing ad-hoc networks, *IEEE Networks Special Issue on Network Security*, 13(6), pp 24 – 30.
- [4] Karlof, C., Sastry, N., and Wagner, D. (2004), Tinysec: A link layer security architecture for wireless sensor networks, in *Proceeding of the 2nd International Conference on Embedded Networked Sensor Systems*, pp. 162–175.
- [5] Karlof, C. and Wagner, D. (2003), Secure routing in wireless sensor networks: Attacks and countermeasures, in *Proceeding of 1st IEEE International Workshop on Sensor Network Protocols and Applications*.
- [6] Hu, Y.-C., Perrig, A., and Johnson, D. B. (2003), Packet leashes: A defense against wormhole attacks in wireless networks, in *Proceeding of IEEE INFOCOM 2003*, vol. 3, pp. 1976 – 1986.
- [7] Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V., and Culler, D. E. (2002), Spins: Security protocols for sensor networks, *Wireless Network Journal (WINE)*, vol. 8, pp. 521–534.
- [8] Wood, A. D. and Stankovic, J. A. (2002), Denial of service in sensor networks, *IEEE Computer*, 35(10), October 2002, pp. 54 – 62.
- [9] Kumar, S. and Spafford, E. H. (1995), A software architecture to support misuse intrusion detection, In *Proceedings of the 18th National Information Security Conference*, pp. 194 – 204.
- [10] Huang, M.-Y., Jasper, R. J., and Wicks, T. M. (1999), A large scale distributed intrusion detection framework based on attack strategy analysis, *Computer Networks*, vol. 31, pp. 2465–2475.
- [11] Ilgun, K. (1993), Ustat: A real-time intrusion detection system for UNIX, In *Proceeding of IEEE Computer Society Symposium on Research in Security and Privacy*, May 1993.
- [12] Ilgun, K., Kemmerer, R. A., and Porras, P. (1995), State transition analysis: A rule-based intrusion detection approach, *IEEE Transactions on Software Engineering*, vol. 21, pp. 181–199.
- [13] Paxson, V. (1998), Bro: A system for detecting network intruders in real-time, In *Proceeding of 7th USENIX Security Symposium*, San Antonio, Texas.
- [14] Porras, P. A. and Neumann, P. G. (1997), Emerald: Event monitoring enabling responses to anomalous live disturbances, In *Proceeding of 20th NIST-NCSC National Information Systems Security Conference*, pp. 353–365.
- [15] Huang, Y. an and Lee, W. (2003), A cooperative intrusion detection system for ad-hoc networks, In *Proceeding of the 1st ACM Workshop on Security of Ad-hoc & Sensor Networks*, pp. 135–147.
- [16] Pires, W. R., Figueiredo, T. H. P., Wong, H. C., and Loureiro, A. A. F. (2004), Malicious node detection in wireless sensor networks, In *18th International Parallel and Distributed Processing Symposium*.
- [17] Marti, S., Giuli, T. J., Lai, K., and Baker, M. (2000), Mitigating routing misbehavior in mobile ad-hoc networks, In *Mobile Computing and Networking*, pp. 255–265.
- [18] Deng, J., Han, R., and Mishra, S. (2003), A performance evaluation of intrusion-tolerant routing in wireless sensor networks, In *Proceeding of IEEE 2nd International Workshop on Information Processing in Sensor Networks*, April 2003, pp. 349–364.
- [19] Buczak, A., Jamalabad, V. (1998), Self-organization of a heterogeneous sensor network by genetic algorithms, *Intelligent Engineering Systems Through Artificial Neural Networks*, C.H. Dagli, et. al. (eds.), Vol. 8, ASME Press.
- [20] Lin, C., Gerla, M. (1997), Adaptive clustering for mobile wireless networks, *IEEE Journal on Selected Areas of Communications*, 15 (7).
- [21] Mathew, R. and Younis, M. (2003), Energy-Efficient Bootstrapping Protocol for Sensor Network, In the *Proceedings of the International Conference on Wireless Networks (ICWN'02)*, Las Vegas, Nevada, June 2003, pp. 287 – 293.
- [22] Gupta, G., Younis, M. (2003), Load-Balanced Clustering in Wireless Sensor Networks, In the *Proceedings of the International Conference on Communication (ICC 2003)*, Anchorage, Alaska, May 2003.
- [23] Gupta, G., Younis, M. (2003), Fault-Tolerant Clustering of Wireless Sensor Networks, In the *Proceedings of the IEEE Wireless Communication and Networks Conference (WCNC 2003)*, vol.3, New Orleans, Louisiana, March 2003, pp. 1579 – 1584.