Detecting Denial of Service Attacks in Sensor Networks

Gu Hsin Lai and Chia-Mei Chen[*]

Department of Information Management
National Sun Yat-Sen University
Kaohsiung 804 Taiwan

Email:cchen@mail.nsysu.edu.tw

Received 16 November 2007; Revised 30 November 2007; Accepted 9 December 2007

Abstract. A wireless sensor network consists of tiny sensing nodes which are deployed in a remote or hostile region, such as battlefield or volcano, to gather information. Each sensor has sensing and wireless communication capabilities, which enable it to gather information from environment and send the information to the remote base station. The applications of wireless sensor networks are used widely today in battlefield monitor, circumstance monitor, or traffic analysis. Security in sensor networks is vital, because sensor networks often apply to a mission-critical task. Therefore, keeping the network available for its intended use is very important. DoS (Denial of Service) attack is that attackers try to diminish or eliminate a network's capacity to perform its expected function. Attackers may simply send vast redundant data to exhaust the resource of a sensor node or drop data to disturb the result of a query. Due to the limited capability of a sensor, it is very difficult to prevent a sensor from DoS attack. Furthermore, mechanisms of DoS detection used in the wired or wireless environments may not be suitable for sensor networks. This paper proposed a cluster-based intrusion detection system. A secure monitor called gNode is proposed to observe and to report DoS attack activities. Each cluster contains a number of gNodes and normal sensor nodes. gNodes send back the warning ticket to its cluster head or sink, if an abnormal event happens. Once a cluster head or sink receives a certain rate of warning tickets if the compromised node is common sensor node, its cluster head would ignore all messages sent from the compromised sensor node, if the compromised node is cluster head, sink would send a re-cluster command to the cluster which its cluster head is compromised. The proposed approach could detect DoS attacks more precisely and reduce the damage from a DoS attack based on clustering. This study expects to establish an energy-efficiency and effective intrusion detection system to detect DoS attacks in wireless sensor networks.

Keywords: sensor networks, Denial of Service, clustering, intrusion detection

1 Introduction

Advances in hardware and wireless network technologies have placed a doorstep of a new era, where inexpensive small wireless devices could provide access to information anytime and anywhere as well as actively participate in creating smart environments. One application for the smart environments uses sensor networks, where the networks are formed by a set of small sensor devices deployed in an ad hoc fashion and sensing a physical phenomenon [1]. Fig. 1 illustrates the environment of a sensor network, where the sensor reports the movement of an object to the sink. Sensors may send data constantly (e.g. the monitor of temperature, air pressure or humidity) or based on occurrence of some events (e.g. the monitor of tank). In this paper, our system is suitable for the former situation.

Sensor networks often carry on mission-critical tasks, such as battlefield awareness, infrastructure protection, or habit monitoring. Surveillance sensor networks in a museum can keep the safety of priceless art crafts from burglaries; similar networks in forest alert when temperature arises abnormally. Therefore, the security of sensor networks becomes very important. Incorrect or unavailable query results may cause serious damage.

A DoS attack is characterized by an explicit attempt by preventing a legitimate user from using a service [2]. Therefore, a DoS attack in sensor network may overload or disable the network, resulting in network performance degradation or incorrect query results.

For applications in sensor networks, loss of availability, DoS attack, may have serious impacts. Loss of availability may cause the failure of detecting a potential accident and result in catastrophic disaster in a factory safety monitoring, loss of availability may leave a back door for enemy invasion in a battlefield surveillance. However, Sensor networks are much more vulnerable to DoS attack than the conventional networks due to the limited capability of sensors and the lack of centralized monitoring and management in sensor networks.

[*] Correspondence author

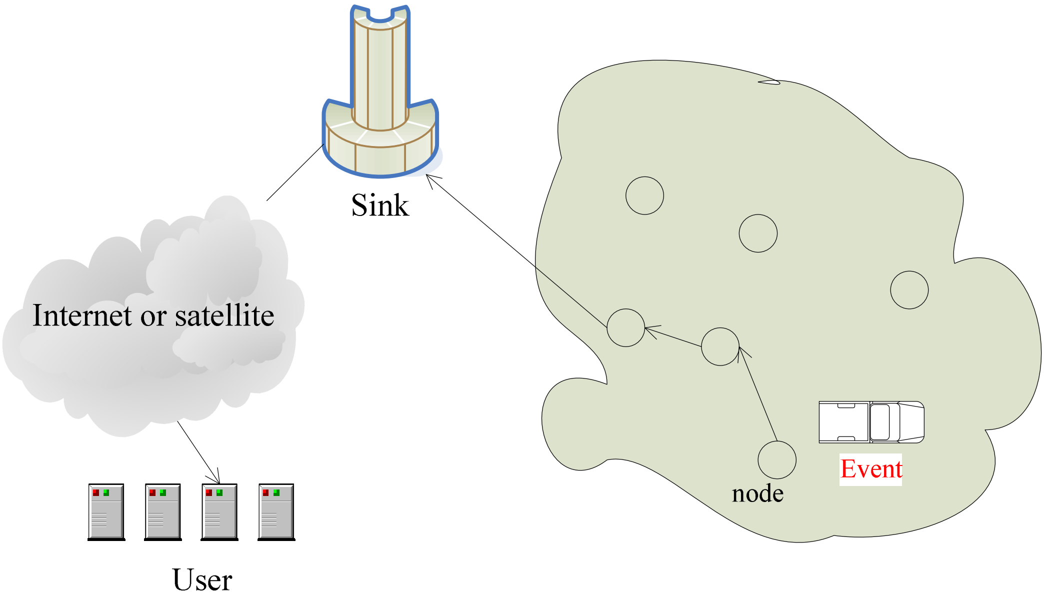


Fig. 1. The environment of a sensor network

Intrusion detection for wireless sensor networks is different from that for Internet or wireless networks. There are two challenges on detecting DoS attacks in sensor networks. Firstly, the limited hardware capability of sensor devices makes the system hard to maintain several long log files or to use comprehensive approaches to analyze the log files. Secondly, the distributed feature of sensor networks makes the system hard to collect overall security related information and analyze the possible anomaly.

In cluster-based sensor network, a cluster consists of several sensors closing to one another. Within a cluster, a special sensor is elected as a cluster head which may have most residual power or lowest communication cost. Once some information is sensed by sensors within cluster, the sensors send this information to cluster head, and cluster head forward them to next hop (next cluster head) or sink. Cluster-based sensor network allow only cluster head (have most residual power or lowest communication cost) to do long distance communication. In addition, cluster head can perform data aggregation or data fusion to reduce the frequency of communication.

Intrusion detection based on cluster approach has several advantages. Firstly, it can reduce the computing cost of IDS. In cluster environment, IDS on only has to log and analyze the traffic flow of cluster head rather than all sensor nodes. Thus, a longer log file can be maintained and the analysis can be more precise. Secondly, in a well-defined clustering algorithm, the cluster head always has more residual battery power, it suitable for some additional security operation and computation. Thirdly, it can minimize the damage caused from DoS attack. In cluster environment, only the cluster head performs the routing operation, if attack is detected by IDS, IDS can inform the attack event to the cluster head, then the cluster head may ignores all message from the adverse node. Thus the damage is minimized. For the above reasons, our approach is based on cluster approach.

In this paper, we deploy a set of special nodes called gNodes (stands for “security guard nodes”) in the network. Instead of performing sensing task, gNodes analyze the network traffic and detect some uncommon behaviors. In this paper, gNodes are deployed to a clustered sensor network, performing detection task only, but not sensing or sending data to other nodes for location hiding purpose. Each cluster may contain a number of gNodes and normal sensor nodes. gNodes send back the warning tickets to the corresponding cluster head or sink. If a compromised sensor vast sends redundant data to its cluster head constantly, gNodes detect this misbehavior and send warning tickets to its cluster head. Once the cluster head receives a certain rate of warning tickets, it would ignore the information sent from the compromised sensor. If a compromised cluster head sends vast redundant data to sink or drop the data received from other cluster head or sensors, gNodes detect this misbehavior and send the warning tickets to sink. Once sink receives a certain rate of warning tickets, it would send the re-elect command to the cluster which its cluster head is compromised and make these sensor nodes elect a new cluster head. Thus the damage could be minimized. In our system, simply statistic approach is used to analyze the traffic data of cluster head and it can minimize the system resource. In addition, based on cluster technology, our system has high detection rate and it can also minimize the damage caused by attackers.

2 Related Works

DoS attack is any event that diminishes or eliminates a network’s capacity to perform its expected function. Hardware failures, software bugs, resource exhaustion, environmental conditions, or any complicated interact on between these factors can cause a DoS [3]. Wood and Stankovic surveyed the DoS attack in different network layers and proposed some defense approaches against the attacks [3].

Huang and Lee developed a learning-based algorithm for automatically computing anomaly detection models based on the correlations among a large set of features. In addition to, they also proposed a cluster-based detection scheme where a cluster of neighboring sensor nodes can periodically, randomly and fairly elect a monitoring node for the entire neighborhood. Huang and Lee's research did not take advantage of the characteristic of cluster technology. Cluster-based sensor network simplified the routing protocol and it makes the detection of DoS easier. Huang and Lee's cluster head is elected to monitor the misbehavior within cluster rather than performing routing and data aggregation operation. In addition, the election of cluster head needs extra cost.

Silva et al. proposed a decentralized IDS model tied to the WSN (wireless sensor network) restrictions and peculiarities; a high-level methodology to construct a specific IDS to a target sensor network with well defined applications; the assessment of the IDS efficiency and accuracy in detecting seven different kinds of simulated attacks [4]. In Silva's research, they defined several rules for detecting misbehavior. A monitor node might have to maintain several types of log files and monitors all of its neighbor nodes. It seems that it's impossible for monitor nodes to hold several long-term log files and it may reduce the performance of IDS.

Rao and Kesidis proposed a traffic transmission patterns be selected to facilitate verification by a receiver. Such traffic patterns are used in concert with suboptimal MAC that preserves the statistical regularity from hop to hop [5]. Their research only considered the situation that compromised sensors drop messages randomly, not taking into account compromised sensors sending vast redundant messages.

Tseng et al. proposed a solution based on specification-based intrusion detection to detect attacks on AODV. Briefly, this approach involves the use of finite state machines for specifying correct AODV routing behavior and distributed network monitors for detecting run-time violation of the specifications [6]. Tseng's research focused on AODV (Ad hoc On-Demand Vector) routing protocol, This research focused on the vulnerable fields in AODV packets and did not consider the situation that the compromised sensors send vast redundant messages or drop messages randomly.

Marti et al. used a watchdog that identified misbehavior nodes and a pathrater that helps routing protocols avoid these nodes [7]. This research did not consider the situation that compromised sensors send vast redundant messages. Marti's watchdog copy every packets that neighbor sensors send to check if compromised sensors drop packets. This approach is not efficient because watchdog needs more memory. If watchdog's neighbor sensors send large number of messages, the watchdog will run out of its memory quickly.

Most researches design sensor node for both sensing and detection function [7,8]. Such design may results in complicated sensor devices. The more complicated a sensor node is, the larger the size and the cost of a sensor node increases, the high possibility the sensor may fail. On the other hand, if every node can perform both sensing and detection simultaneously, a compromised node may send false detection information to the sink and achieve the attack purpose easily. It is hard for a sensor node to distinguish whether the information sent from a sensor is genuine or fake. The load of each sensor increases, as one needs to perform both sensing and detection the network life time may become shorter. Thus, the proposed solution introduces gNodes to perform detection operation to solve this problem.

Few researches studied intrusion detection on cluster-based sensor networks. In Huang and Lee's clustered-based solution, a cluster head is responsible for monitoring misbehavior rather than performing routing or data aggregating operation. This kind of cluster can't take advantage of the cluster technology (routing or data aggregating). Furthermore, since every node may have an opportunity of being a cluster head, every node needs more memory space, computing capability and battery power to be able to perform detection tasks. On another point of view, the proposed solution might malfunction if an attacker compromises a cluster head or a compromised node may declare to be a cluster head. A more novel solution is required to solve the above mentioned problems.

3 Sensor Network Model

In this paper, we assume that (1) in a cluster, only cluster head can perform routing operation, (2) there are three types of nodes in the sensor network -- cluster head, sensor node, and gNode and all possess same capability, (3) any nodes (sensor node. cluster head, or gNode) may be compromised, (4) two types of attack may occur (greedy and neglected), where greedy attack makes the victim sends redundant data and neglect attack suppresses the data, (5) discrete time is used in the system, (6) sensors send data at stable transmission rate.

The proposed cluster-based intrusion detection system in a sensor network environment is illustrated in Fig. 2. A set of special nodes called "guarding nodes" (gNodes) are deployed in the sensor network. Instead of performing the common sensing task, gNodes analyze the network traffic and detect abnormal network transmission behaviors.

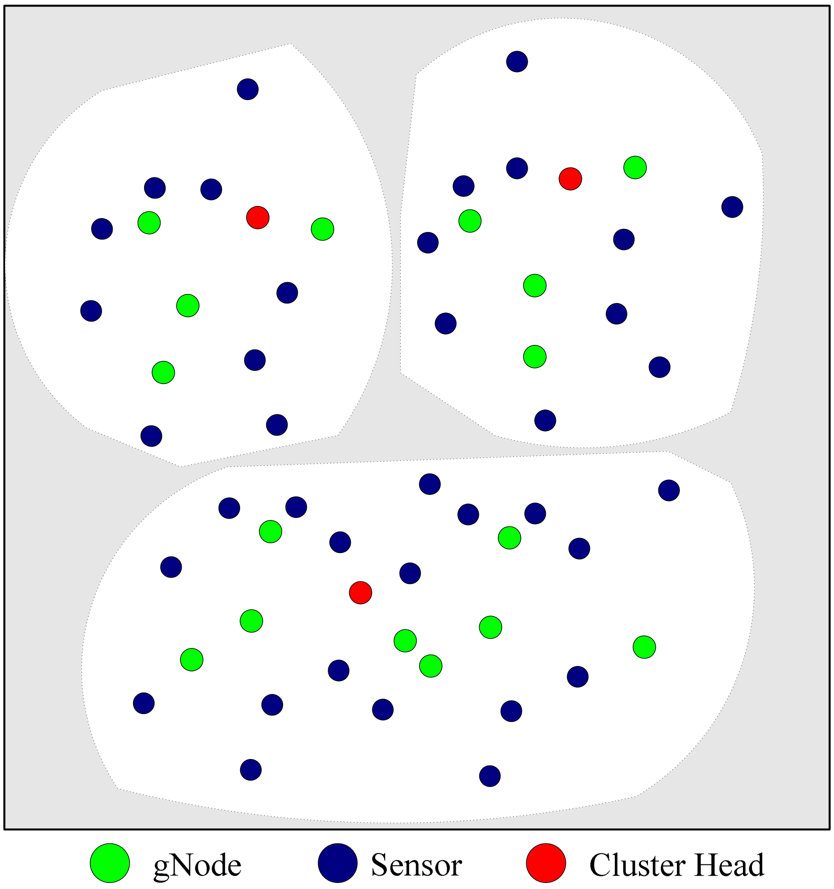


Fig. 2. System model

Cluster construction of the proposed sensor network could be any methods proposed by the previous researchers, such as HEED [9] or LEACH [10]. In the cluster head selection mechanism, the chosen cluster head broadcasts the declared message within its cluster when it is selected as a cluster head. gNode then responses to the cluster head when it receives the message like sensors. Therefore, each cluster contains a number of gNodes and normal sensors and each cluster head is aware of which and how many gNodes are in its cluster.

In the proposed system, each gNode only monitor the flow traffic of its cluster head, where flow is the total data volume transmitted. Therefore, each gNodes only needs to maintain single history flow information, instead of large or multiple log files. A gNode records inbound/outbound flow traffic to/from the cluster head and computes the current flow profile, such as the average flow and its standard deviation. gNodes could detect the misbehavior based on the flow profiles.

4 The Proposed Approach

In the proposed system, there are three types of nodes, namely cluster head, sensor node, and gNode and all may be compromised. As mentioned in the previous section, two types of DoS attacks may compromise the network. A greedy node would send vast redundant data, while a neglect node suppresses the data. Different DoS attack on different node type may cause different impact to the network. Hence, the proposed solution adopts different approaches to tackle the attacks. The detection approach and the action after detection for different attack type on different node type are summarized in Table 1.

Table 1. The summary of the proposed detections and actions

Node type	Attack type	Detection	Action
sensor	greedy	Detected by the cluster head when receiving certain number of warning tickets from gNodes.	The cluster head ignore the messages from the compromised sensor.
sensor	neglect	N/A	N/A
cluster head	greedy	Detected by the sink when receiving certain number of warning tickets.	The sink sends re-elect message to the cluster.
cluster head	neglect	Detected by the sink when receiving certain number of warning tickets.	The sink sends re-elect message to the cluster.
gNode	greedy	Detected by the cluster head or the sink when receiving at least two same warning tickets in a time interval.	The greedy gNode cannot cause damage as the warning rate would not exceed the predefined threshold.
gNode	neglect	N/A	N/A

To describe the detail of the approach, the following notations are needed and hence described below.

1. S_{id}: The ID of a sensor node.
2. C_{id}: The ID of a cluster head.
3. M_{cid}: The number of messages which cluster head C_{id} sends to other cluster heads or the sink.
4. M_{sid}: The number of messages which sensor S_{id} sends to its cluster head.
5. N_{cid}: The number of sensors in the cluster whose cluster head is C_{id}.
6. G_{cid}: The number of gNodes in the cluster whose cluster head is C_{id}.
7. C_{avg}: The average number of messages which the sensors in a cluster send to the cluster head.
8. C_{std}: The standard deviation of the number of messages which the sensors in a cluster send to the cluster head.
9. C_{upper}: The upper bound of the number of messages that a sensor would send. For example, gNode considers an abnormal situation when $M_{sid} > C_{upper}$[1].
10. C_{lower}: The lower bound of the number of messages that a sensor would send. For example, an abnormal behavior might occur the average transmitted messages is less than the lower bound, i.e., $M_{cid} < C_{lower}$[2].
11. E_{sid}: the count of continuous misbehaviors of sensor S_{id} that a gNode observes.
12. E_{cid}: the count of continuous misbehaviors of cluster head C_{id}.
13. R_{sid}: The number of gNodes reporting misbehavior of sensor S_{id}.
14. R_{cid}: The number of gNodes reporting misbehavior of cluster head C_{id}.
15. T_{cot}: The threshold of misbehavior count. A node is compromised if its misbehaviors exceed the threshold.
16. T_{ratio}: The threshold for the ratio of the number of warning tickets to the numbers of gNodes in a cluster. Higher ratio means higher number of warning tickets fined to a node and higher possibility that it is compromised.

Detecting a compromised sensor

Suppose that a sensor is attacked. Only greedy attack would be a threat to the network. Sensors are not reliable and might be failed due to the lack of power or other problems. Therefore, if a sensor drops some messages and does not transmit data, the cluster head or sink would aggregate the rest of the collected data and response the

[1] C_{upper} is configurable, for example, $C_{upper}= C_{avg} +\alpha^* C_{std}$.

[2] C_{lower} is configurable, for example, $C_{lower} = C_{avg} -\beta^* C_{std}$.

requested query. A neglect sensor would be considered as unreliable in the network. A greedy sensor may send vast redundant data continually to the cluster head as illustrated in Fig. 3. When the inbound flow of the cluster head exceeds the upper bound of the flow volume, i.e., $M_{sid} > C_{upper}$, gNodes monitoring the cluster head record the abnormal flow phenomenon on the cluster head. Then, gNodes start to monitor flow volume from each sensor to the cluster head. A greedy sensor would exceeds the upper bound of the flow volume for a sensor (i.e., $E_{sid} > T_{cot}$) and gNodes would send warning tickets to the cluster head.

If gNodes in the cluster would be able to detect the abnormal misbehavior, they send the warning ticket to the cluster head. The cluster head considers that the sensor is compromised and discards the messages when a certain fraction of gNodes in the cluster believes it is compromised. ($if\ \frac{\sum R_{sid}}{G_{cid}} > T_{ratio}$).

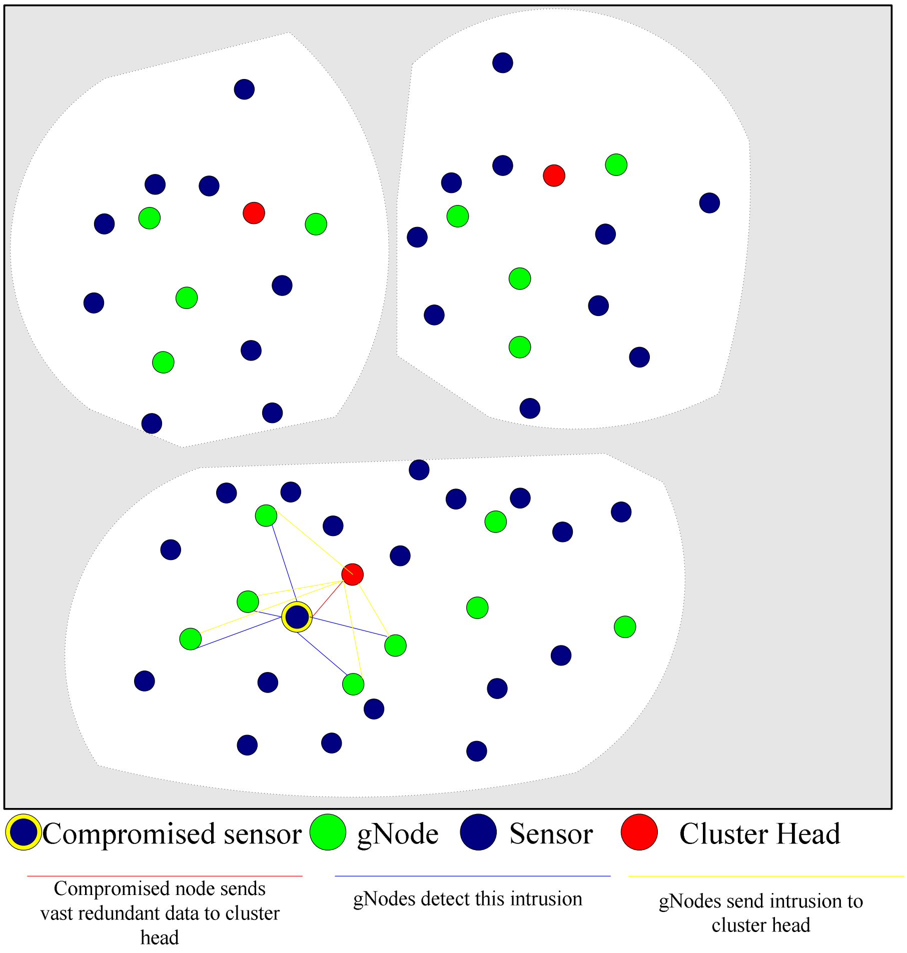

Compromised sensor
gNode
Sensor
Cluster Head
Compromised node sends vast redundant data to cluster head
gNodes detect this intrusion
gNodes send intrusion to cluster head

Fig. 3. An illustration of a compromised sensor detected by gNodes

Detecting a compromised cluster head

A cluster head carries major functions, such as aggregating and routing data. A cluster head is vulnerable to both greedy and neglect DoS attack. A compromised cluster head may drop data or retransmit redundant data. Either case results in abnormal flow volume of the cluster head. Therefore, by monitoring both inbound and outbound flow volume of the cluster head, gNode would be able to detect such abnormal flow, either the outbound flow is lower than the expected lower bound ($M_{cid} < C_{lower}$) or the outbound flow exceeds the upper bound ($E_{cid} > T_{cot}$). A warning ticket fined to the cluster head would send to the sink as illustrated in Fig. 4.

As the sink collects a certain fraction of warning tickets ($\frac{\sum R_{cid}}{G_{cid}} > T_{ratio}$), it sends a re-cluster message to the compromised cluster to reselect a new cluster head.

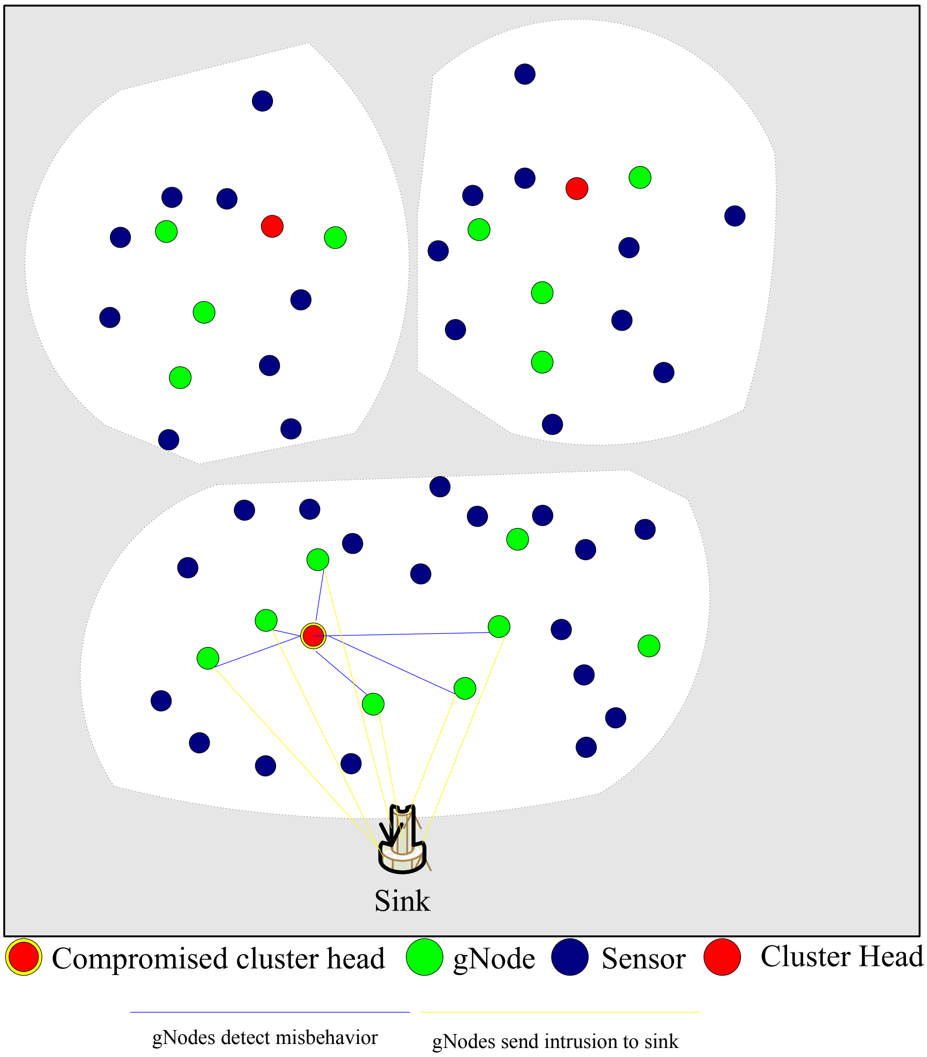


Fig. 4. An illustration of a compromised cluster head

Detecting a compromised gNode

Similar to regular sensors, neglect attack to gNode is not a threat to the system. If a compromised gNode keeps sending incorrect or redundant warning tickets to the cluster head, the cluster head only count them one as they are sent from the same gNode. Therefore, the warning ticket rate (the ratio of the number of reported warning tickets from distinct gNodes to the total number of gNodes in the cluster) would not exceed the predefined threshold (i.e., $\frac{\sum R_{cid}}{G_{cid}} < T_{ratio}$) and no damage caused by the compromised gNode. The cluster head then would ignore all the warning tickets from the compromised gNode.

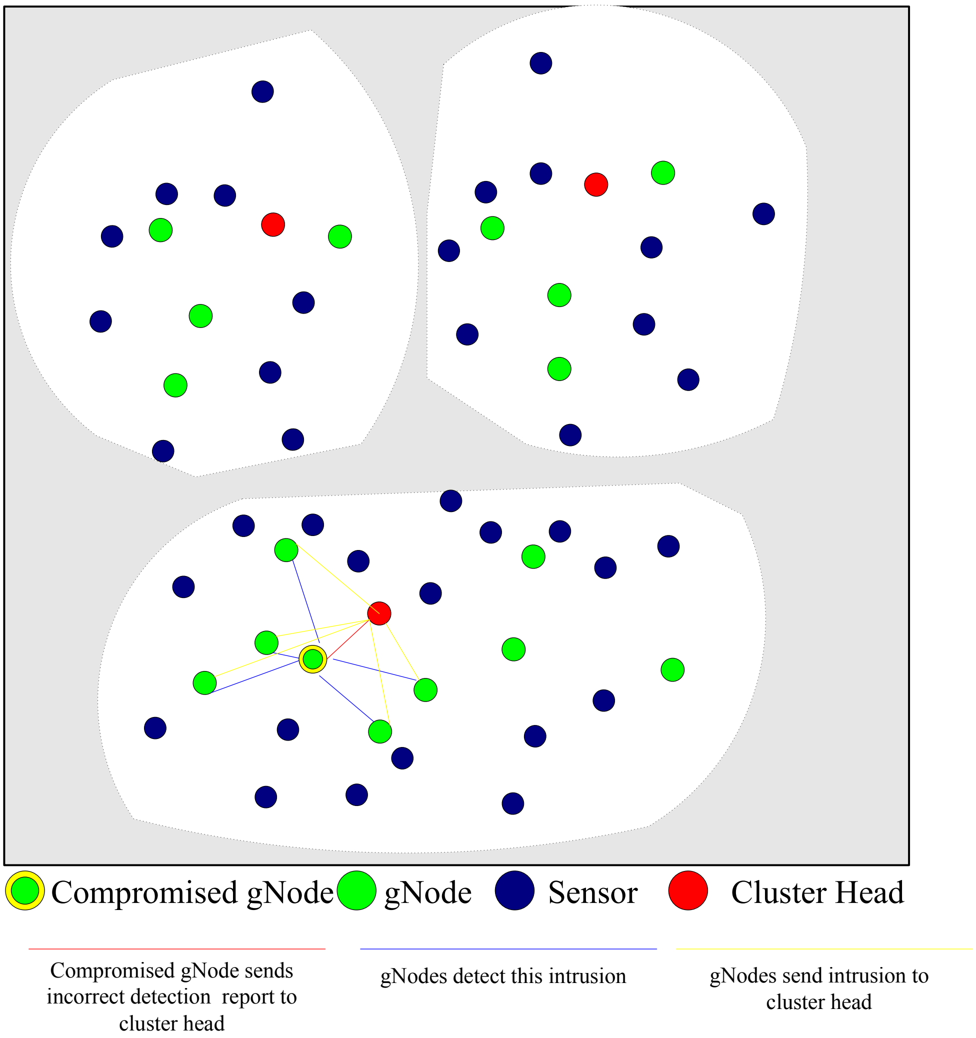
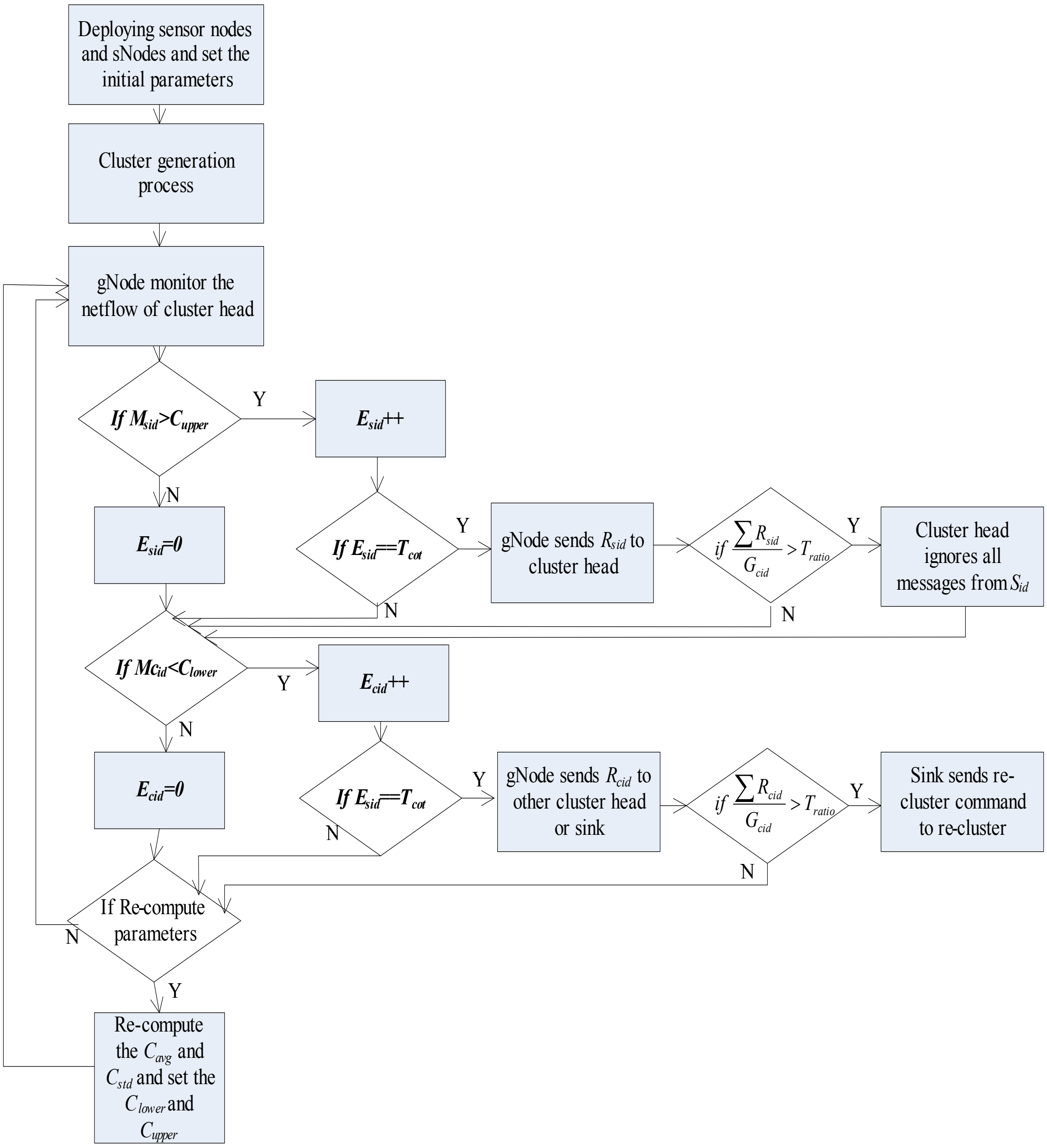

Compromised gNode
gNode
Sensor
Cluster Head
Compromised gNode sends incorrect detection report to cluster head
gNodes detect this intrusion
gNodes send intrusion to cluster head

Fig. 5. An illustration of a compromised gNode

The advantage of the proposed approach is cluster-based and the damage and detection can be done in the scope of a cluster. gNodes could get averaged flow profile data and keep a log file only for their own cluster. The proposed detection method only uses simple statistic analysis for detection. The complete system process is shown in Fig. 6.



Deploying sensor nodes and sNodes and set the initial parameters
Cluster generation process
gNode monitor the netflow of cluster head
If M_{sid}>C_{upper}
Y
E_{sid}++
N
E_{sid}=0
If E_{sid}==T_{cot}
Y
gNode sends R_{sid} to cluster head
$if\ \frac{\sum R_{sid}}{G_{cid}} > T_{ratio}$
Y
Cluster head ignores all messages from S_{id}
N
N
If Mc_{id}<C_{lower}
Y
E_{cid}++
N
E_{cid}=0
If E_{sid}==T_{cot}
Y
N
gNode sends R_{cid} to other cluster head or sink
$if\ \frac{\sum R_{cid}}{G_{cid}} > T_{ratio}$
Y
Sink sends re-cluster command to re-cluster
N
If Re-compute parameters
N
Y
Re-compute the C_{avg} and C_{std} and set the C_{lower} and C_{upper}

Fig. 6. System process

5 Performance Evaluations

Simulation is conducted for evaluating the performance of the proposed method. The system performance is measured by the detection rate and false alarm rate. Clustering is generated by LEACH algorithm with the probability of 0.05 which a sensor node becomes the cluster head. 90 gNodes are deployed in the simulated systems in all experiments except Experiment 5. Each experiment generates 100 simulated environments and averages the results.

In the simulation, we assume that the message transmission of each node is independent and is generated by a Poisson distribution. Let λ_s be the mean of the normal transmission rate, λ_a be the mean of the transmission rate of a greedy node, and $\lambda_{a.}> \lambda_s$. A negelect node would drop a message with the propability of D_p. Let N be the total number of nodes in the network. Every a period of time, T time units, each gNode re-computes the average message volume and its standard deviation, C_{avg} and C_{std}. Let the whole system life last for R periods.

Table 2 illustrates the parameter setting of Experiment 1. Fig. 7 shows the results of Experiment 1 with greedy attack and Fig. 8 for neglected attack.

Table 2. The parameter setting of Experiment 1.

Parameter name	Value
T_{cot}	3
D_p	0.5
$\lambda_{a.}$	10
G	90
N	400
T	400
R	10
$F_{ratio.}$	0.001
C_{upper}	C_{avg} +2* C_{std}
C_{lower}	C_{avg} -2* C_{std}

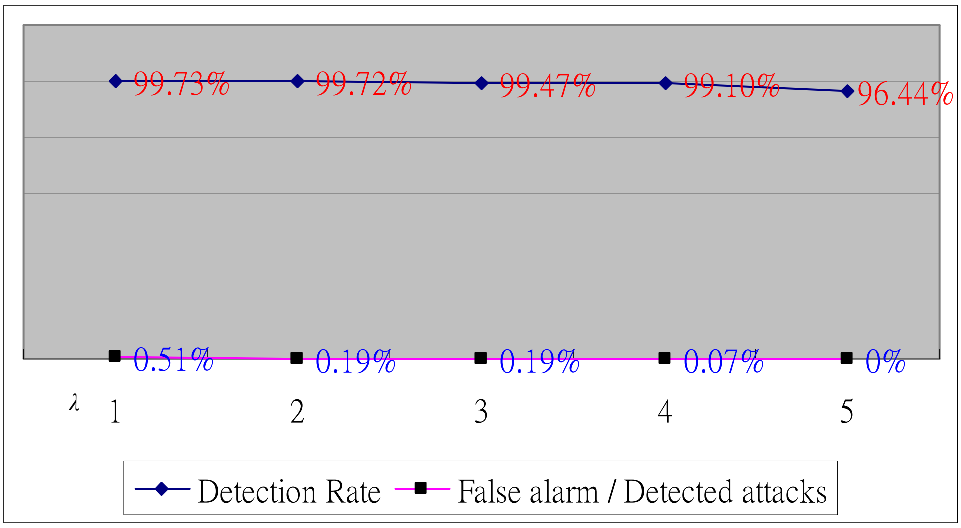


Fig. 7. The results of Experiment 1 with greedy attack.

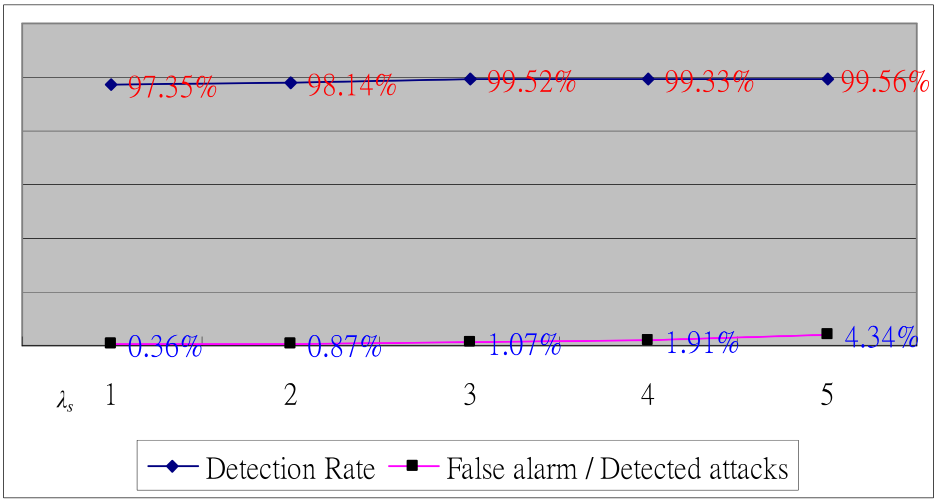


Fig. 8. The results of Experiment 1 with neglected attack.

Based on the results of Experiment 1, we can observe that our system has higher detection ratio and lower false alarm. Even when the attack rate is approaching to the transmission rate (λ_a is close to λ_s , λ_s =5, λ_a =10), the detection ratio is still very high (greedy: 96.44%, neglected: 99.56%). The results of Experiment 1 also indicate that the false alarm of neglected attack increases if λ_a is close to λ_s. However, even when the attack rate is approaching to the transmission rate (λ_s =5, λ_a =10), the false alarm / detected attacks is still very low (4.34%).

Table 3. The parameter setting of Experiment 2.

Parameter name	Value
λ_s	3
D_p	0.5
$\lambda_{a.}$	10
N	400
T	400
R	10
F_{ratio}.	0.001
C_{upper}	C_{avg} +2.5* C_{std}
C_{lower}	C_{avg} -2* C_{std}

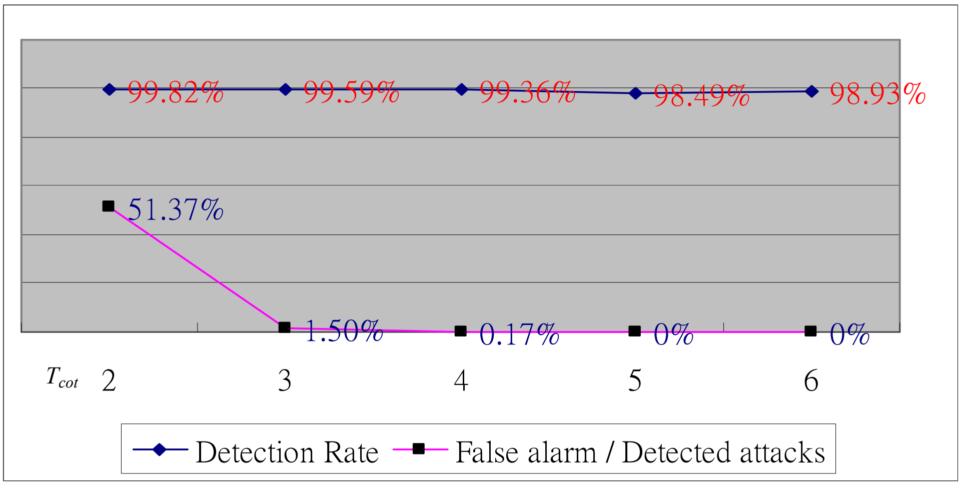


Fig. 9. The results of Experiment 2 which attack type is greedy.

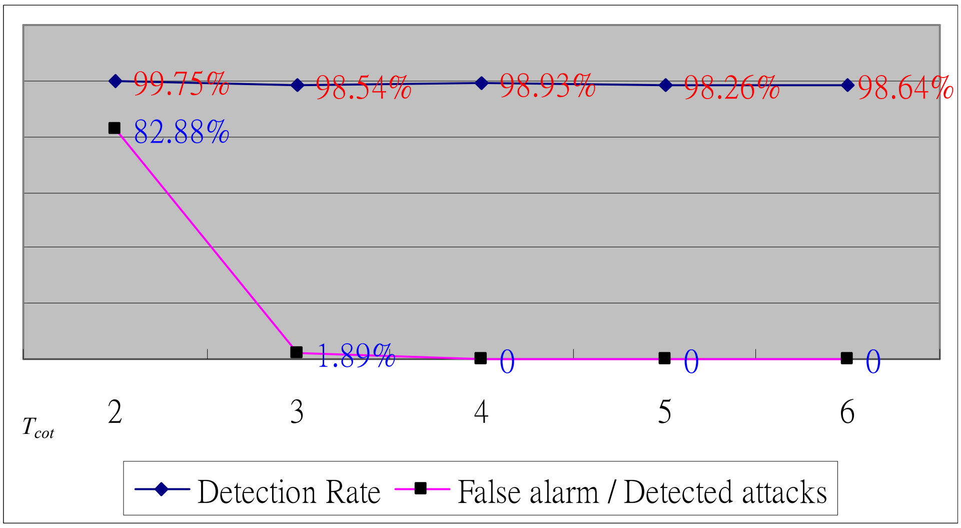


Fig. 10. The results of Experiment 2 which attack type is neglected.

Table 3 presents the parameter setting of Experiment 2. Fig. 9 shows the results of Experiment 2 greedy attack and Fig. 10 for neglected attack.

The results of Experiment 2 indicate that T_{cot} is a critical parameter; if the value of T_{cot} is too low, the number of false alarms will increase. In Experiment 2, we can see that the false alarm/ detected attacks is high (greedy:51.37%, neglected:82.88%), when the value of T_{cot} is set to 2 and that the number of false alarms becomes very small once T_{cot} is larger than 2. Therefore, T_{cot} can be set to a value higher than 2 for reducing false alarms. Theoretically, the higher T_{cot} is, the lower the detection ratio the system possesses. In addition, based on the results, we can see that the detection rate is steady and high (greedy: 98.93%, neglected: 98.64%) and is not sensitive to the change of T_{cot}.

Table 4. The parameter setting of Experiment 3.

Parameter name	Value
λ_s	3
D_p	0.5
$T_{cot.}$	3
N	400
T	400
R	10
F_{ratio}.	0.001
C_{upper}	C_{avg} +2.5* C_{std}
C_{lower}	C_{avg} -2* C_{std}

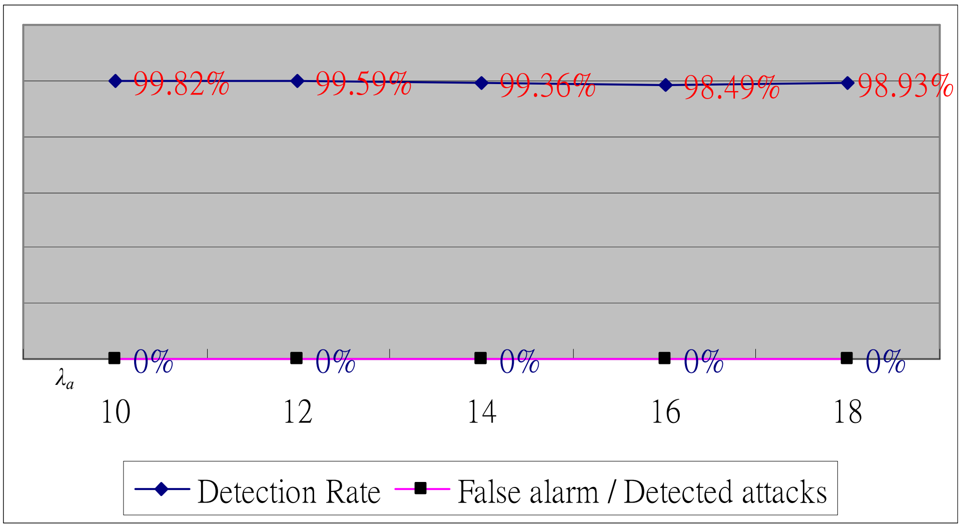


Fig. 11. The results of Experiment 3 with greedy attack.

The parameter setting of Experiment 3 with greedy attack is described in Table 4 and the results is shown in Fig. 11. The system performs steady well. We can observe that the higher the λ_a the higher the detection ratio of greedy attack. In real world environment, the value of λ_a is always much larger than λ_e, it means that in real world environment, our system has very high detection ratio of greedy attack.

Table 5. The parameter setting of Experiment 4.

Parameter name	Value
λ_e	3
λ_a	10
$T_{cot.}$	3
N	400
T	400
R	10
F_{ratio}.	0.001
C_{upper}	C_{avg} +2.5* C_{std}
C_{lower}	C_{avg} -2* C_{std}

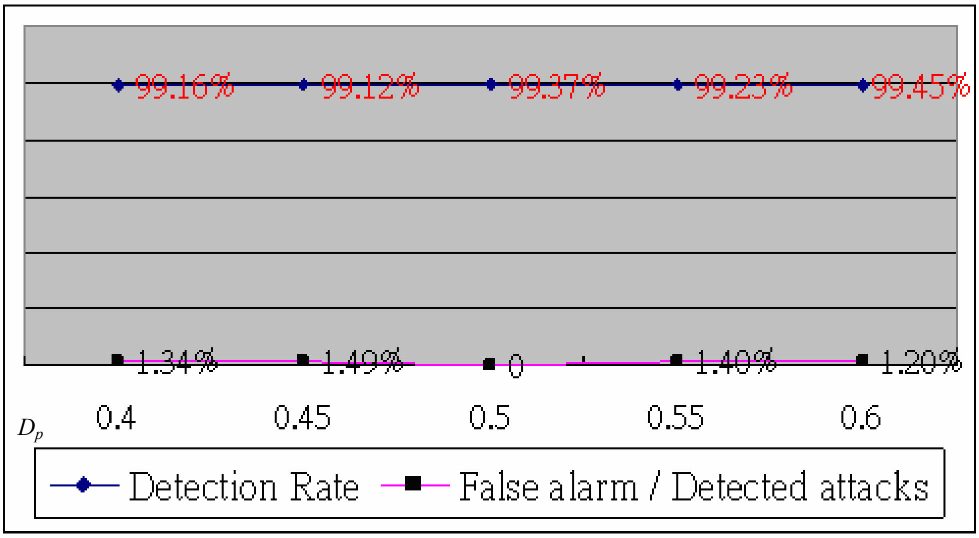


Fig. 12. The results of Experiment 4 which attack type is neglected.

Table 5 illustrates the parameter setting of Experiment 4. Fig. 12 illustrates the results of Experiment 4 which attack type is neglect. The results of Experiment 4 illustrate that the higher the D_p the higher the detection ratio of neglected attack and it also shows that the performance of our system is very good. The detection ratio is larger than 99%. In real world environment, the value of D_p is never too small (if D_p is too small, the influence is also small), it means that in real world environment, our system has very high detection ratio of neglected attack.

Table 6. The parameter setting of Experiment 5.

Parameter name	Value
λ_e	3
λ_a	10
$T_{cot.}$	3
N	400
T	400
R	10
F_{ratio}.	0.001
C_{upper}	C_{avg} +2.5* C_{std}
C_{lower}	C_{avg} -2* C_{std}

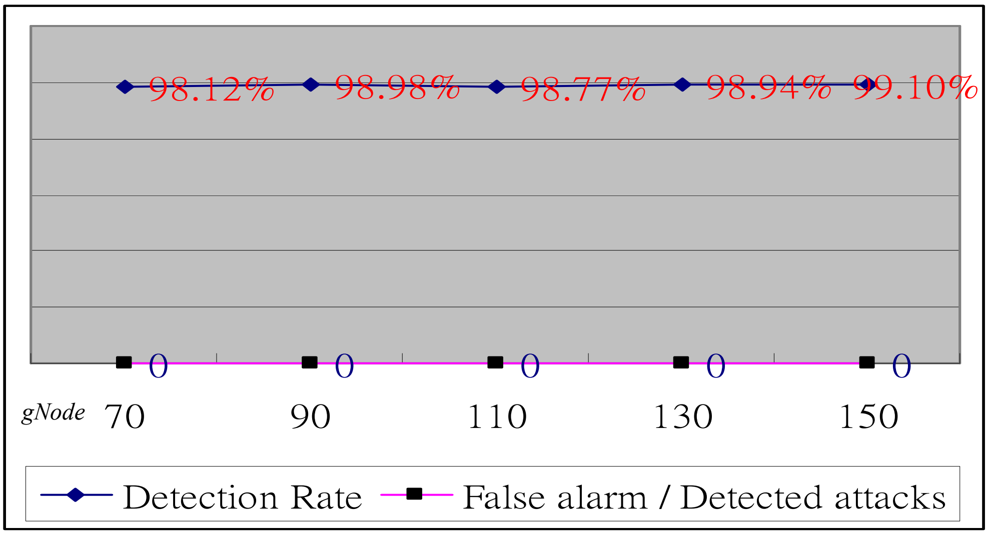


Fig. 13. The results of Experiment 5 which attack type is greedy

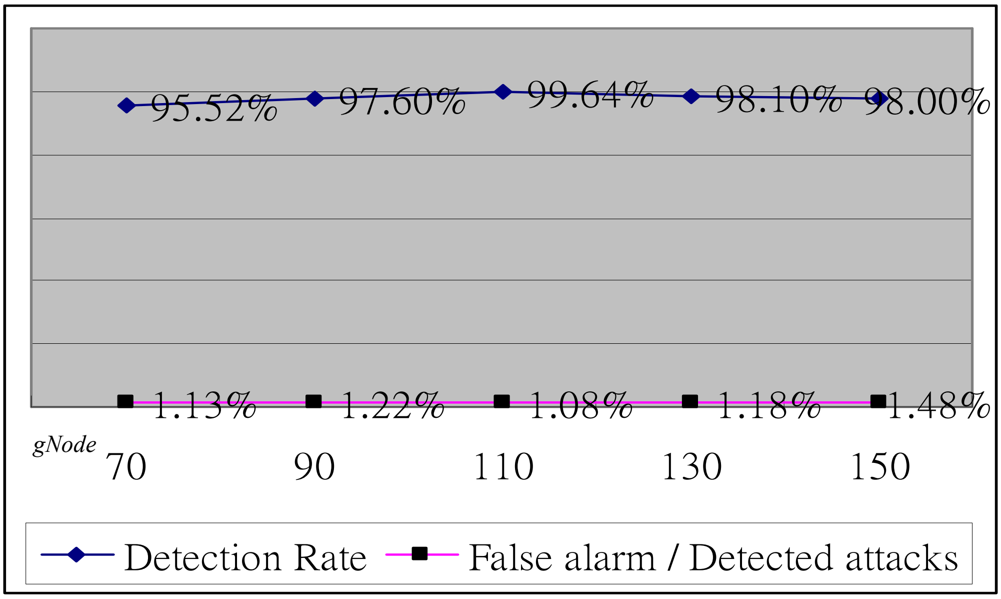


Fig. 14. The results of Experiment 5 which attack type is neglect

Table 6 illustrates the parameter setting of Experiment 5. Fig. 13 illustrates the results of Experiment 5 which attack type is greedy and Fig. 14 illustrates the results of Experiment 5 which attack type is neglected. In our system, the number of gNodes may affect the performance. Theoretically, the more gNodes means the higher performance but higher costs. Experiment 5 shows that our system has higher detection rate even only few gNodes are deployed in our system. It also means that the cost of our system is small.

The results of all the experiments show that the proposed system has very high detection ratio and low false alarm rate. Therefore, we can conclude that our detection method performs very well.

6 Conclusions

In this paper, we propose a distributed, cluster-based IDS is proposed to prevent sensor network form DoS attack. In our system, a set of special nodes called "security guard nodes" (gNodes) are deployed in the sensor network to gather the statistical data of cluster head. The proposed approach has several advantages: (1) load balancing can be achieved based on clustering; (2) damage caused by a DoS attack may be minimized if the network is clustered; (3) the detection is more reliable, since the cluster head determines a compromised node based on a number of warning tickets, instead of one warning ticket; (3) if a single gNode is compromised or captured, it can not affect the network by sending wrong warning ticket. Only when a large number of gNodes are compromised simultaneously, which would happen rarely, the network may be affected. Before multiple gNodes are compromised, the attack action may be detected beforehand; (4) the proposed approach could achieve information hiding, since a gNode only sends warning ticket to its cluster head or sink. Hence, it lowers the probability of being compromised or captured; (5) common sensors do not perform detection and hence the battery power is conserved.

In our system, we need to guarantee that the confidentiality and integrity of the reports which are sent from gNodes. And the future work may focus on the key management of gNodes. In addition to, what if there are no gNodes within a cluster? To solve this problem, a deployment mechanism may be designed to solve this problem. This problem might be solved by other method. In a cluster, sensor nodes will become gNode with a probability. This method ensures there are some gNodes in a cluster. Therefore, key management and deployment of gNodes may be our future work.

Acknowledgement

This work was supported in part by TWISC@NCKU, National Science Council under the Grant NSC 96-2219-E-006-009.

References

[1] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, “A taxonomy of wireless micro-sensor network models,” *ACM SIGMOBILE Mobile Computing and Communications Review*, Vol.6, No.2, pp 28-36, 2002.

[2] Computer Emergency Response Team/Coordination Center, “Denial of Service Attacks,” *http://www.cert.org/tech_tips/denial_of_service.html*, 2001.

[3] A. D. Wood and J. A. Stankovic, “Denial of service in sensor networks,” *IEEE Computer*, Vol.35, No.10, pp.54-62, 2002.

[4] Y.-A. Huang and W. Lee, “A cooperative intrusion detection system for ad hoc networks,” *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, Virginia, pp.135-147, 2003.

[5] R. Rao and G. Kesidis, “Detecting malicious packet dropping using statistically regular traffic patterns in multihop wireless networks that are not bandwidth limited,” *Proceedings of the IEEE Global Telecommunications Conference,* Vol.5, pp.2957-2961, 2003.

[6] C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt, “Intrusion detection: A specification-based intrusion detection system for AODV,” *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pp.125-134, 2003.

[7] S. Marti, T. J. Giuli, K. Lai, and M. Baker, “Mitigating Routing Misbehavior in Mobile Ad Hoc Networks,” *Proceedings of the 6th annual international conference on Mobile computing and networking*, Boston, pp.255-265, 2000.

[8] A. Paula, R. Silva, M. H. T. Martins, B. P. S. Rocha, A. A. F. Loureiro, L. B. Ruiz, and H. C. Wong, “Decentralized intrusion detection in wireless sensor networks,” *Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks,* Montreal, Canada, pp.16-23, 2005.

[9] O. Younis and S. Fahmy, “HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” *IEEE Transactions on Mobile Computing,* Vol. 3, No.4, pp.366-379, 2004.

[10] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks,” *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, January 4-7, Hawaii, Vol.2 pp.1-10, 2000.