Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 8961239, 11 pages
https://doi.org/10.1155/2018/8961239

Research Article

A Security Framework for Cluster-Based Wireless Sensor Networks against the Selfishness Problem

Zeba Ishaq, Seongjin Park, and Younghwan Yoo

Department of Electrical and Computer Engineering, Pusan National University, Busan, Republic of Korea

Correspondence should be addressed to Younghwan Yoo; ymomo@pusan.ac.kr

Received 3 March 2018; Revised 30 May 2018; Accepted 26 June 2018; Published 10 July 2018

Academic Editor: Houbing Song

Over the last few decades, Cluster-Based Wireless Sensor Networks (CBWSNs) have played a crucial role in handling various challenges (load balancing, routing, network lifetime, etc.) of large scale Wireless Sensor Networks (WSNs). However, the security becomes a big problem for CBWSNs, especially when nodes in the cluster selfishly behave, e.g., not forwarding other nodes' data, to save their limited resources. This may make the cluster obsolete, even destroying the network. Thus, a way to guarantee the secure and consistent clusters is needed for proper working of CBWSNs. We showed that the selfishness attack, i.e., passive attack or insider attack, in CBWSNs can cause severe performance disaster, when particularly a cluster head node becomes selfish. In order to prevent this situation, this paper proposes a security framework that involves a novel clustering technique as well as a reputation system at nodes for controlling selfishness, making them cooperative and honest. The novelty of the clustering comes from the existence of inspector node (IN) to monitor the cluster head (CH) and its special working style. The experimental results showed that the proposed security framework can control the selfishness and improve the security of the clusters.

1. Introduction

The recent advances in sensing and communication capabilities of WSNs have made a wide range of applications possible, which can be divided into two main categories: tracking and monitoring. In order to meet the requests from various applications including military, habitat, health, business, public, industrial, and environmental ones, WSNs are developed into more expertized systems like terrestrial, underground, underwater, and multimedia WSNs [1, 2]. WSNs based applications need to deploy a large number of sensor nodes over phenomenal environment, and all those sensor nodes send sensing data to a sink node; thus, a lot of congestion and data collisions can occur in WSNs. This will result in the depletion of limited energy from the network in a short time. In all these circumstances, node clustering can address these issues because it can provide load balancing and efficient resource utilization [3–5]. In other words, clustering is indispensable for scalability and network lifetime extension.

In addition to the energy efficiency and scalability of clustering architecture itself, each cluster must be secure and reliable, but it has not gained much attention so far. In a cluster, the cluster head plays an important role in aggregating and forwarding data sensed by other nodes. Thus, malfunctioning or compromisation of CH can lead to unreliable data delivery. Then besides the malfunctioning and compromisation, the selfishness of a node can be a significant problem for the cluster network. Let us assume that a node thinks that its battery energy is the most valuable resource and that it decides not to forward others' packets to save the energy. If the node takes the CH role, the problem becomes more serious. This is called the selfishness attack or passive attack because it can harm the network even though the selfish node has no explicit intention to attack the network. Thus, there should be a cluster leader election protocol that is more efficient, resilient, and effective than previous techniques [6–8].

The previous research in the security domain revealed that the inside attacks by the authorized nodes are far more difficult to be controlled than the outside attacks by the unauthorized nodes. The selfishness attack is one of the critical inside attacks. Various methods have been suggested to control this type of inside attacks [9, 10]. First, incentive

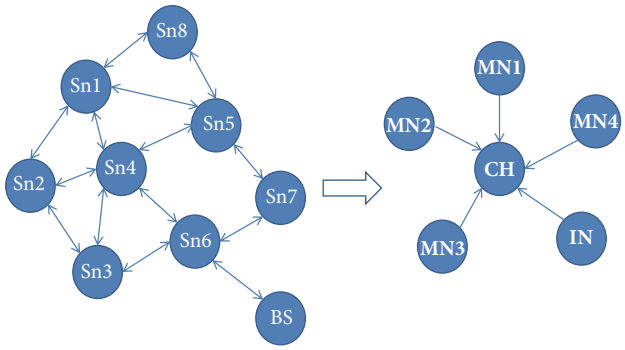

Figure 1: WSN: before and after cluster formation.

schemes have been suggested to resolve the selfishness attack, which encourage the nodes to be honest by giving some credits when they participate in a cooperative environment (e.g., MANETs) [11, 12].

On the other hand, reputation and trust systems punish selfish nodes by giving them penalties of bad reputation, finally resulting in the exclusion from the network. These reputation systems are useful for any system to avoid being a victim of inside attacks. These days, a trust system and a reputation scheme are very important for the wireless communication [13–15].

Our motivation is to mitigate the selfishness problem in CBWSNs in order to provide robust clusters to maximize the lifetime of network. This prompted us to propose a security framework to fight against this inside attack. The main design of the proposed security framework is to appoint two special nodes per cluster: inspector node and cluster head node. The resulting cluster then basically consists of three types of nodes, i.e., CH, IN, and MNs (member nodes); and they are one hop away from CH as shown in Figure 1. In order to control the selfishness attack, these nodes act in a special way and an additional security is provided by using a reputation system at each node. The IN exploits the packet overhearing scheme, which is one of the characteristics of wireless communication and used by many previous researches to provide security against the selfishness attack [16, 17].

If an IN finds some problems while overhearing CH's transmission, then it blacklists the CH and also informs MNs within its range to make them stop forwarding data to the CH. However, the MNs also can refuse the IN's decision if they judge, based on their own reputation system, it might be a deliberate accusation by the IN. Meanwhile, CH also sends random checking requests to IN, to ascertain its status, whether IN is working correctly or not. In addition, CH assigns bad reputation values to MNs which do not take part in IN nomination for a long time to save their energy. The main responsibility of CH is to forward MNs' data to the sink node. The proposed strategy not only solves the selfishness (passive attack) but also covers some of the active attacks including the black hole, the selective-forwarding, the on/off, and the transmission opportunity-wasting attack by analyzing the overheard data at IN.

This paper improves the previous work in [18], which only focuses on how the proposed method works. On the other hand, this paper investigates how the method performs well through the extensive simulation result and how it can handle various active attacks as well. The rest of the paper is organized as follows. The next section describes the proposed solution in detail. Section 3 compares briefly the proposed security architecture with some existing schemes. Section 4 describes the evaluation. Section 5 discusses the expected outcomes of the proposed solution. Section 6 concludes this paper.

Table 1: Nodes selfishness levels.

Nodes	Partial Selfishness	Full Selfishness	Physical Damage
CH	30 %<PF<100%	PF=<30 %	PF=0
IN	30%<RCR<100%	RCR=<30 %	RCR=0
MNs	30%<Rep<100%	Rep=<30%	Rep=0

2. Proposed Solution

2.1. Selfishness Attack. In our scenario, there are three types of selfish nodes as follows. All these types of selfishness attack should be addressed.

(1) Selfish CH: it drops data packets instead of forwarding to the sink node.

(2) Selfish IN: it stops overhearing CH or sends deliberate accusing messages on CH.

(3) Selfish MNs: it does not properly participate in the CH and IN election process. It means that it does not present itself for the IN nomination and also does not reply to CH election process deliberately.

Moreover, considering typical situations these nodes can behave either fully or partially selfishly. It means that they do not perform their roles continually or intermittently. For example, under partially selfish behavior, the data forwarding of CH, overhearing of IN, and participation of MNs in election process can be stopped intermittently. On the other hand, if these activities are stopped for a long while, then nodes can be considered as fully selfish or dangerous. The intensities or levels of selfishness are the result of the intent of free riding or hiding their selfishness. This also provides the basis for differentiating types of deliberate accusing attacks later. In order to correctly quantify and identify these situations, we made some assumption as shown in Table 1, where *PF* denotes the packet forwarding rate, *RCR* is the rate of reply to random checking, and *Rep* is the rate of reply to a request of a neighbor who volunteers to be CH.

2.2. Description of the Method. Before describing the proposed method, notations are given. The whole network consists of N sensor nodes. The number of neighbors of a sensor node is denoted by M and M is less than N. After the clustering each cluster consists of CH, IN, and MNs. IN and MNs are one hop away from CH. The proposed strategy has three levels as shown in Figure 2, and each level has an important role to fight against the selfishness problem in its own way. The first level involves a uniqueness of clustering

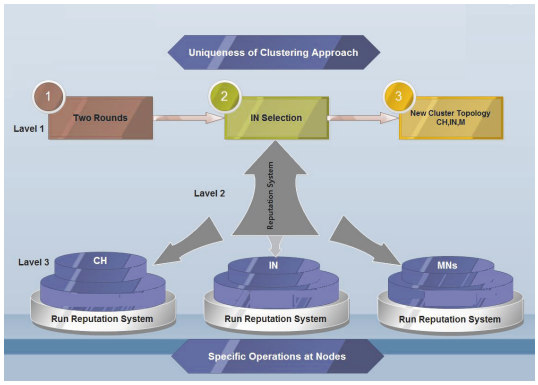

FIGURE 2: Structure of the proposed security framework.

process, which incorporates two special nodes, i.e., CH and IN, per cluster. The clustering process is carried out in two stages. The second level takes the role of the reputation scheme contributing to the other two levels and the third level reveals specific operations at CH, IN, and MNs beside their usual activities. The levels of the proposed security framework will be elaborated one by one in the following section.

2.2.1. Level 1: Uniqueness of Clustering. The proposed clustering process is carried out in distributed manner and has two stages where the first stage is further divided into two rounds for selecting CH and IN.

(i) First Stage: it comprises two rounds. In the first round, a node that wants to be a cluster head broadcasts a request to its one-hop away neighbors. The neighbor's replies depend on their previous experience with the node. It means that they check the previous reputation for the node and confirm it against the predefined threshold reputation value as described in next section. Only trustful nodes can receive the replies from the neighbor nodes. It is the first step towards controlling the selfishness problem, to avoid a selfish node being selected as CH. Next is the confirmation of CH, as we already assumed that every sensor node knows that it has a list of M possible neighbors. Therefore, if the number of replies or replying nodes denoted by Rep satisfies a specific threshold value, T_s, where

$$T_s = \frac{M}{2} \tag{1}$$

it shows that value of threshold should be equal to half of the neighbor nodes. Therefore,

$$Rep >= T_s \tag{2}$$

Thus, the requesting node announces itself as the cluster head and all replying neighbor nodes become the MNs of the cluster as shown in Figure 3. But the situation might not be so straight due to the possibility that a malicious node can pose as CH even though it does not receive enough replies. It just pretends to work as CH not to get a bad reputation from neighbors. But some neighbors have bad reputation on it already due to its uncooperative nature, naturally having not replied to CH selection process. Nonetheless, if they get a membership message from it, then they blacklist it. It means other nodes expel the node from the network. Nodes in the blacklist cannot send their packets through other nodes, nor can they become CHs in the future. On the other hand, if any node is trustworthy and receives enough replies to become CH; then the cluster is formed and the CH has good reputation for future use.

Right after the first round, the second round starts: CH requests only its neighbor MNs to volunteer or the IN role since IN must be within the communication range of CH to overhear its transmission. The volunteering MNs send back replies to CH. Waiting for the MNs replies for specific time duration, CH checks if the number of replies of MNs denoted by Rep_1 satisfies a specific threshold value, T_{s1}:

$$Rep_1 >= T_{s1} \tag{3}$$

In our research, the threshold T_{s1} is set to a half of the neighbor nodes:

$$T_{s1} = \frac{M}{2} \tag{4}$$

Among the nodes sending a reply, CH selects the most reputable node as IN that completes the second round of the first stage. The process is shown in Figure 4.

(ii) Second Stage: in the second stage, actual communication takes place among CH and their MNs. CH forwards data, carries out random checking process, and updates reputation values for its MNs and IN. On the other hand, IN overhears CH transmission, responds to random checking requests of CH, and manages its reputation system data, while MNs send sensing data to CH and maintain its reputation system.

(i) **Design Issue: Why clustering has two stages instead of one?** The debate is that, why using two stages instead of one? In order to monitor the selfishness attack by CH, IN should be one hop away from CH, or within the range it can overhear CH's transmission consistently. Then, if sensor nodes that want to be CH and IN present themselves in the same stage, there is a chance for the two selected nodes to be out of the range of each other. This contradicts with the requirement that CH and IN must be the direct neighbor of each other. This situation is illustrated in Figure 5.

2.2.2. Level 2: Reputation System. The reputation and incentive schemes have been used for many years against the inside attacks. There are many conventional reputation systems in use that were specially designed for social and e-commerce applications. Most of them have centralized and complicated structure and consider many parameters (direct and indirect observations) as metrics for consistent trust and reputation computations. We also assumed that every node in the network has a reputation system and an initial reputation value for every one hop away neighbor. It has a distributed and very simple design that frequently considers direct observation (overhearing of data forwarding activity) like many other methods in MANET that care for data forwarding activity or route request reply as metrics to decide reputation for their nodes and occasionally uses indirect observation

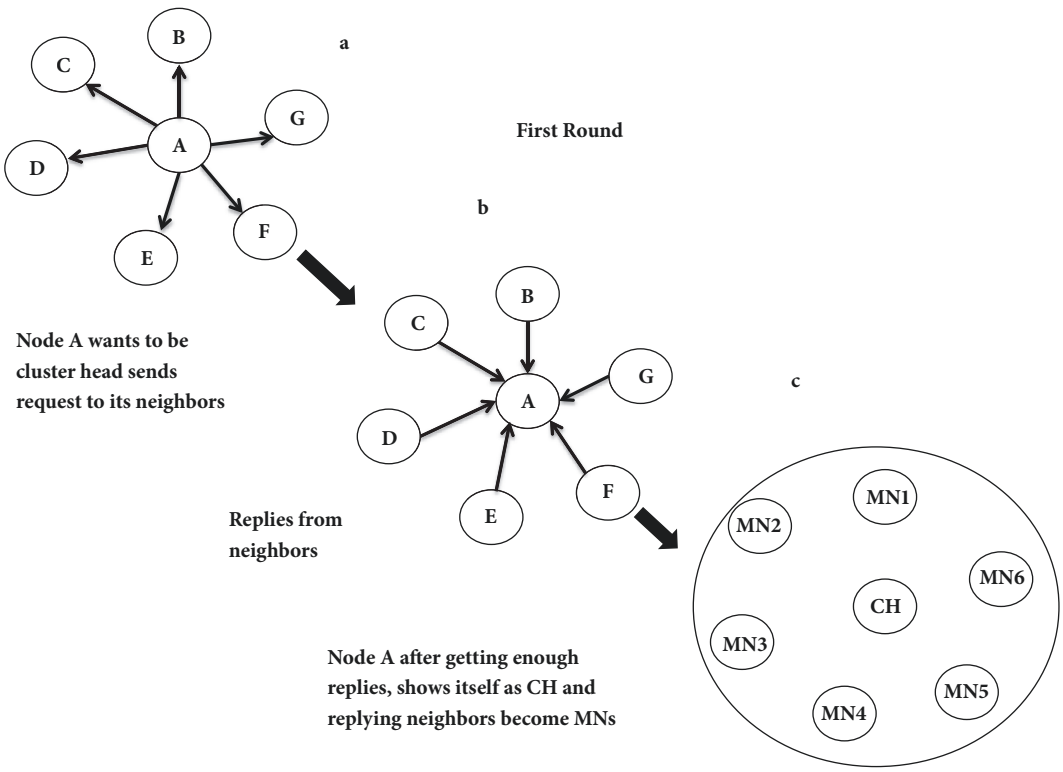

Figure 3: First round of first stage of the proposed cluster establishment.

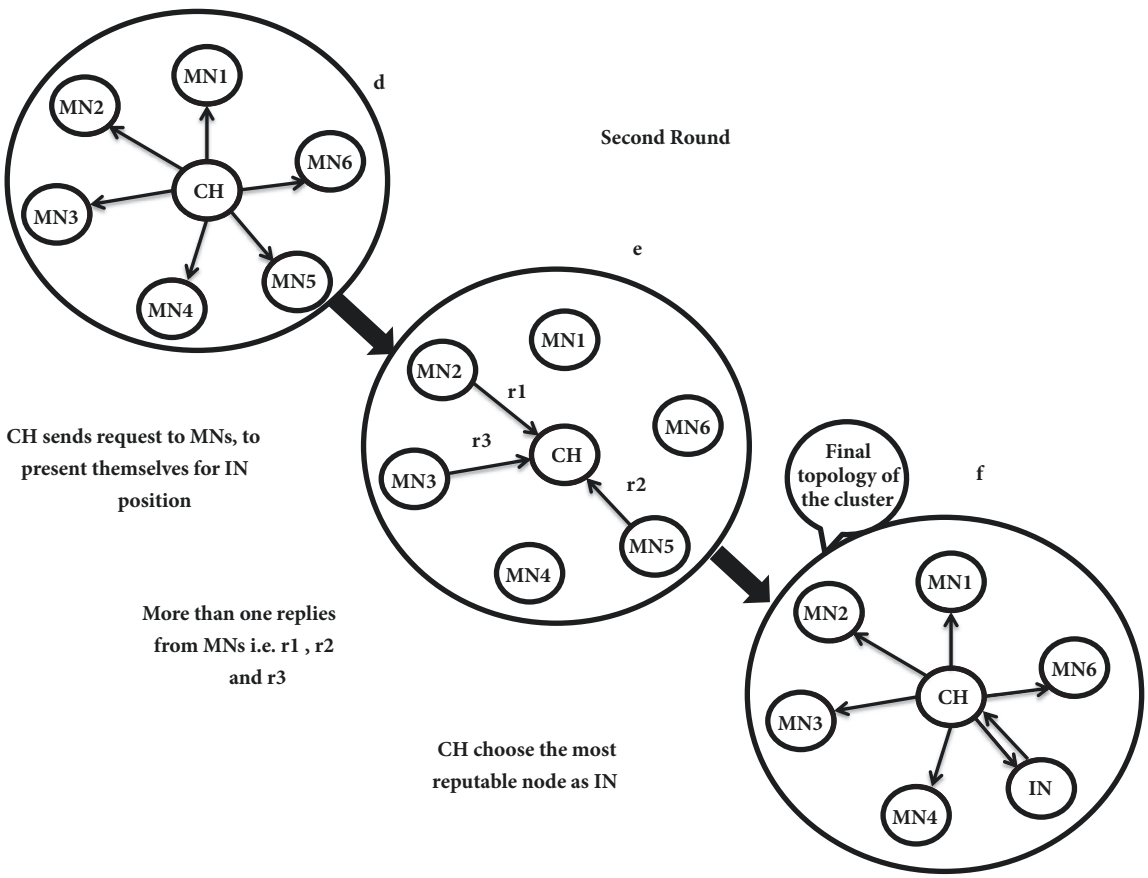

Figure 4: Second round of first stage of the proposed cluster establishment.

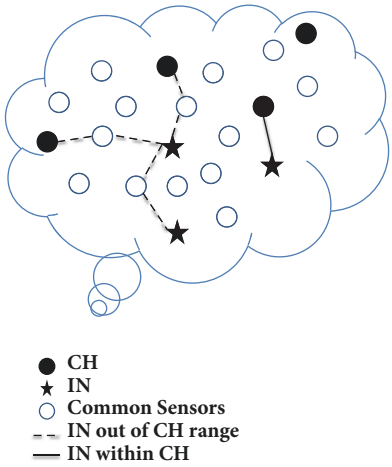

Figure 5: Possible scenarios for CH-IN relations when they are elected in the same stage.

Table 2: Neighbors of nodes.

Nodes	Neighbors
CH	IN, MNs
IN	CH, MNs
MNs	CH, MNs, (IN)

for reputation values calculation as will be discussed in active deliberate accusation scenario in this section. These reputation values are changed depending on how cooperative they are in data forwarding activity and later can also be used to elect an honest node as CH in the clustering process as mentioned above. We made some assumptions related to these reputation values before cluster formation as follows:

$$IR\ \left(Initial\ Reputation\right) = R \tag{5}$$

$$UF\left(Updating\ F\ actor\right) = \pm 1 \tag{6}$$

where R is an integer; e.g., if *R=3* then *IR=3*, and it increases or decreases by ± 1, according to their behavior in data forwarding. In Figure 6, we can see that node A updates its neighbor node B's reputation depending on its behavior as described above.

After cluster formation, a network is divided into clusters with three types of nodes, i.e., CH, IN, and MNs as shown in Figure 1. The neighbor types of each node may be a little different depending on its role in the cluster as listed in Table 2. Note that MNs may or may not be within the communication range of IN depending on the position inside the cluster. Maintaining reputation history, the reputation scheme at each node operates in a little different way depending on its role in the cluster. First, CH can increase or decrease reputation value to IN depending on its response to the random checking process and can also evaluate reputation values of MNs based on their participation in the IN election process. Second, IN evaluates the reputation value of CH by overhearing whether it properly forwards packets coming from MNs. Lastly, MNs also evaluate CH based on the delivery ratio of their own packets to the destination. Based on this reputation values MN can make its own decision whether it can still trust CH or not, when IN accuses the CH selfishness. In this way, we can avoid the deliberate accusation problem on CH by IN. Moreover, again the continuation of their activities can matter for comparing accumulated reputation values of these nodes against the predefined threshold to be free of partial selfishness.

(i) Deliberate Accusation: two types of deliberate accusation are possible, active, or passive deliberate accusation.

(a) Active deliberate accusation: a malicious IN may purposely accuse CH though it is working honestly and has good reputation. In this situation, MNs can play a vital role in making their own decision using the CH reputation they have collected for themselves. The deliberate accusation can be restrained by giving the bad reputation to this kind of IN and notifying it to the CH. CH will take any further action like a new IN election.

(b) Passive deliberate accusation: this type of deliberate accusation can appear in two situations. First, some neighbors of a node that want to be a CH may not reply to the request to volunteer CH, regardless of its previous reputation. Second, some neighbors of a specific node do not volunteer for IN election only when the node takes the role of CH. The reason behind not sending the reply in the both cases can be to save their energy. In other words, they just behave selfishly.

(ii) **Control of Deliberate Accusation:** the deliberate accusation on CH is harmful not only for the CH but also for the entire network since the total number of nodes participating in the network is reduced. Thus, this should be controlled. In case of the active deliberate accusation, MNs as well as CH can take action against selfish IN according to their previous reputation values. MNs notify this situation to CH and CH can assign bad reputation values for this IN and further can elect a new IN as described above. On the other hand, the passive deliberate accusation can appear in two situations. First, it can badly affect the clustering process; clustering either takes a long time to complete or even may fail. In order to prevent this situation, we can use the facts that we already assumed; i.e., every node in the network has a reputation history for its one hop away neighbors, *M* is the list of possible neighbors, and *Rep* is the list of expected replying nodes that should satisfy threshold T_s as given (1) and (2); then *m,* the list of neighbors that have bad reputation value for the requesting node due to its noncooperative behavior in the past, can be calculated as follows:

$$m = \left(M - Rep\right) \tag{7}$$

However, in case of a passive deliberate accusation scenario where some nodes of requesting node may not reply to the request to volunteer CH, we assume that the new list of expected replying nodes will be Rep_2; then *N*, the combined list of neighbors that either deliberately accuse or have bad

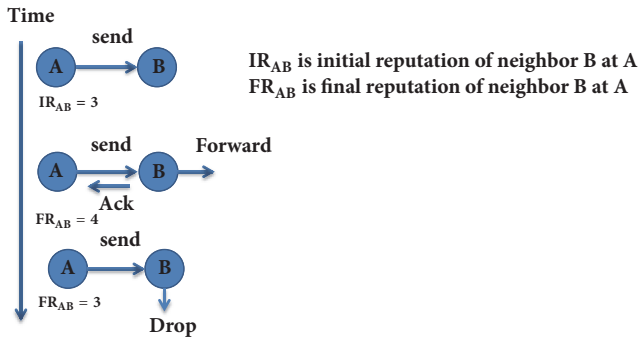

Figure 6: Reputation system operations.

reputation for the requesting node due to its noncooperative behavior, can be calculated as follows:

$$N = (M - Rep_2) \tag{8}$$

where N is greater than m; then we can find deliberate accusing nodes as follows:

$$N_{DA} = (N - m) \tag{9}$$

where N_{DA} denotes the number of deliberately accusing neighbors and faulty nodes during the replying process of the CH election and M_{new} is the new list of possible neighbors.

$$M_{new} = (M - N_{DA}) \tag{10}$$

In the second case, the problem can be solved by CH by picking up most reputable neighbor and assigning the IN role to it. If CH observes that some MNs do not present themselves for an IN role during the past some period, then it assigns them bad reputation values. This type of deliberate accusation can be harmful since the role of IN is concentrated only to several nodes. In this way, we can say that the reputation schemes can help in controlling deliberate accusation and make the clustering process consistent and reliable.

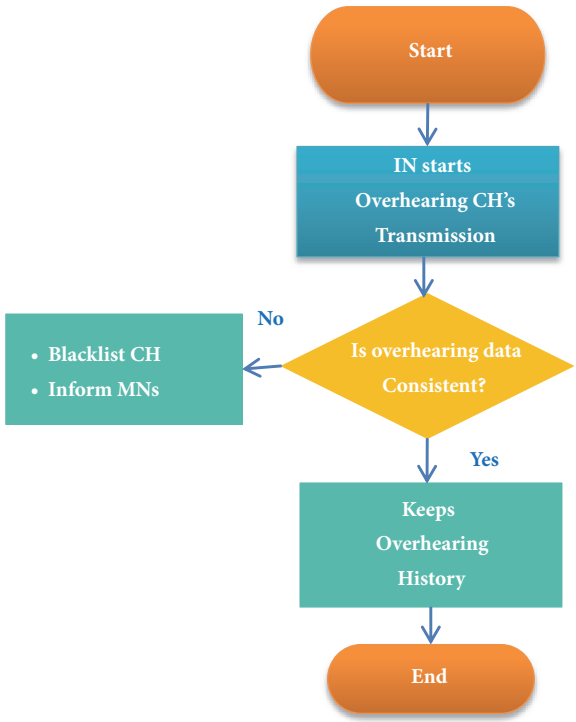

Figure 7: IN's inspection of the CH's selfishness problem.

2.2.3. Level 3: Operations at Specific Nodes

(i) **Inspector Node (IN):**

(a) Transmission overhearing: as the basic function of CH is to forward the packets on behalf of MNs, the selfishness of CH is very dangerous to the cluster, even destroying the whole cluster. In order to avoid this situation, IN can play a vital role. Overhearing the transmission of CH, if IN observes that CH does not forward more than a fixed number of packets, it sends an accusation message to MNs in the same cluster to induce them to put the CH in the blacklist. If any node is blacklisted on the reputation system of others, the cooperation with it is refused by others and even it can be excluded from the network. The workflow of IN is given in Figure 7.

(b) Response to Random Checking by CH: IN may not work correctly after the election, i.e., not overhearing CH's transmission to save its own energy? In order to prevent this situation, CH checks if the IN continuously overhears its transmission or not. CH randomly requests the packets that IN overheard for some fixed duration just before the request time. For the sake of energy efficiency, the random checking process utilizes an information hashing technique. A simple hash function can map data of arbitrary size to data of fixed and relatively small size [19]. Upon a random checking query; IN sends CH the hash values of the requested packets instead of the whole packets. By sharing the same hash function, CH can check if the IN has continuously overheard its transmission or not. The use of hash function

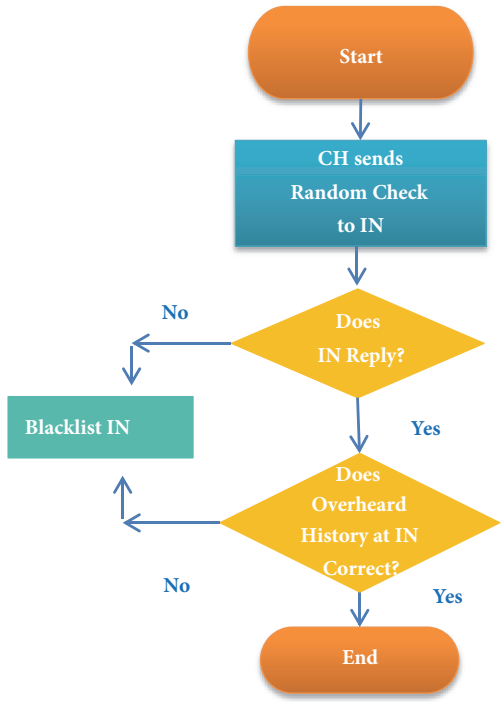

Figure 8: CH's random check for IN's selfishness attack.

definitely lessens the burden of IN and CH in terms memory, communication, computation, and energy consumption. It also provides data integrity. If IN cannot answer correctly the request, IN is accused by CH of its selfish behavior. Therefore, IN must always keep hash values of the transmission overhearing history for the fixed duration in the manner of the sliding window. IN removes all previous history after satisfying each random checking request and starts to keep the overhearing history again to satisfy the next random checking request. In this manner, IN can be refrained from becoming selfish during its working.

(ii) **Cluster Head (CH):** CH also has to handle the selfishness of others. As mentioned earlier, all nodes in the network should willingly play the role of CH or IN for normal operation of the entire network, although more energy needs to be consumed. This is why nodes not playing a role of CH or IN for a long time are blacklisted on a reputation system. However, as mentioned above, a node may avoid this obligation easily by not doing the IN's operation, i.e., the overhearing on CH, after being elected as IN. It just pretends to work as IN, becoming a free rider. As time grows, the number of dishonest INs may increase, and then CH can have much room to behave selfishly without receiving any penalty. To prevent this situation, we make the CH check IN by requesting a specific packet randomly which the IN has overheard. If the IN has been working honestly, it could answer the request correctly. Since IN cannot know in advance which packet will be requested, it cannot cheat the CH random checking process. In case of any malfunctioning of IN, CH blacklists it and notifies other MNs as shown in Figure 8.

(iii) **Member Nodes (MNs):** all sensor nodes have a reputation system to enhance the cooperation of their neighbors. MNs evaluate the reputation of CH and IN. For CH, the reputation is proportional to the rate of its packets being successfully delivered to the destination. On the other hand, for IN, the reputation is based on how similar the decisions of the IN on the reputation of others are to its own reputation system. For example, if an IN accuses a CH, but a MN can judge the CH as a cooperative node based on its own reputation system, the IN is regarded as a deliberate accuser, being assigned bad reputation by the MN. On the contrary, if an IN does not accuse a CH even though a lot of packets are not delivered to the sink node, a MN considers the IN as a selfish node.

3. Comparison with Other Techniques

Here, we concisely compared our security framework against selfishness attack with existing schemes for the security of cluster head election, focusing on the schemes in [20–22]. The common goal of these schemes is to provide security for cluster head node election against active attacks by using various technologies. However, they have several limitations. First, they can handle only active or external attacks, while our security framework can control the selfishness attack (inside attack) as well as several active attacks. Second, they are centralized schemes, using a base station to make a decision about the head nodes. Such centralized approaches are considered costly in terms of communication, computation, and maintenance. Hence, they are not suitable for WSNs having resource constraints. In contrast, our solution is a distributed scheme to avoid the single point of failure and excessive usage of resources. It does not incur that much communication and computation cost and is much more secure than centralized schemes. Third, the three election protocols in [21] use lightweight cryptographic algorithms, but they are susceptible to numerous attacks. Lastly, the protocols in [22] using digital signatures involve considerable computation overhead and are vulnerable to DoS attacks, being not suitable for resource limited tiny WNS nodes. Meanwhile, our scheme adopts a reputation system that is more resilient to the selfishness attack on the cluster head node. Moreover, the use of hash function also makes it more efficient in terms of communication, computation, energy consumption, and memory overhead.

4. Evaluation

In order to evaluate and analyze the proposed framework, we performed simulation using MatLab and other parameters as shown in Table 3. Total 50 nodes are randomly distributed over an area of 500m∗500m and initially consist in nine clusters. The actual distribution of the nodes and clusters is shown in Figure 9. We highlighted only four clusters here in order to avoid confusion and clearly show the effect of our proposed idea.

Figure 10 shows IN's response against selfish CHs. Two CHs (CH: 48 and CH: 45) were discovered to behave selfishly. In other words, they were observed by INs to stop data

Table 3: Simulation parameters.

Parameters	Value
Simulator	Matlab
Area	500∗500, 1000∗1000
Number of nodes	50, 100
Node Deployment	Random
Communication Range	100m
Ratio of Selfish Nodes	10-50

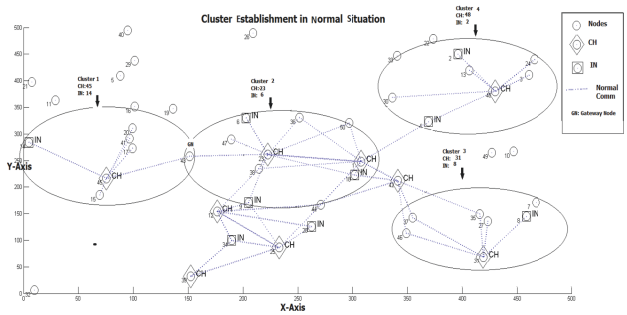

Figure 9: Cluster establishment.

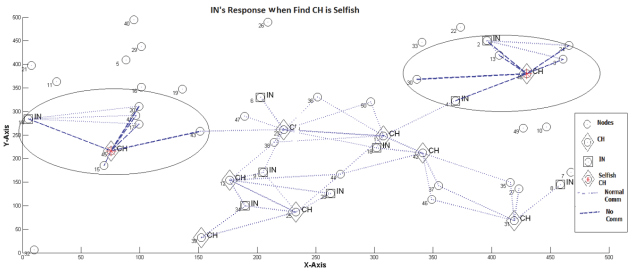

Figure 10: Response of IN against selfish CH.

forwarding for their MNs (MNs: 2, 3, 4, 13, 24, 30 and MNs: 14, 15, 17, 20, and 41. After finding this situation, INs (IN: 2 and IN: 14) informed MNs within their range (3, 13, 24 and 17, 20, 41) to stop communication with the CHs denoted by small dotted line in Figure 10. Then, MNs made their decision according to IN's information as well as their own experience with CHs. As a result, the CHs were put on the blacklist by INs and MNs. The blacklisted CHs were marked with an asterisk inside the circle.

In Figure 11, we showed the CH's response when it found out the selfish IN. For this purpose, we made IN: 8 not properly overhear the CH transmission and even not properly respond to the random checking process of CH. When the IN's reputation at CH fell down the threshold value, the CH blacklisted this IN and started the process to elect a new IN as shown in Figure 11.

We also quantitatively analyzed the performance of the proposed security framework in terms of number of dropped packets while considering the partially and fully selfishness of nodes. The simulation environment was extended, 100 nodes in the range of 1000m∗1000m. The nodes were deployed randomly over an area of interest and the ratio of selfish nodes

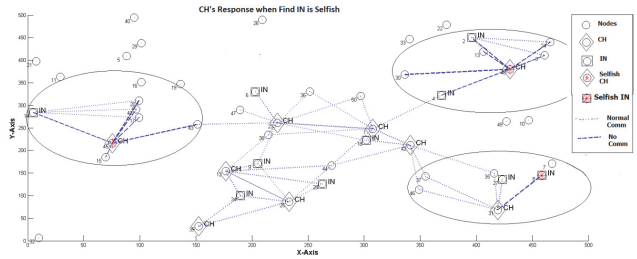

Figure 11: Response of CH against selfish IN.

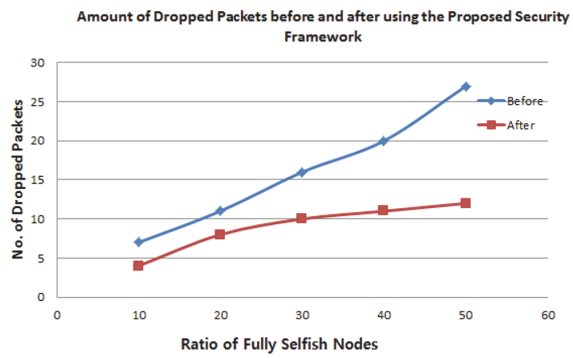

Figure 12: Number of dropped packets while using fully selfish nodes.

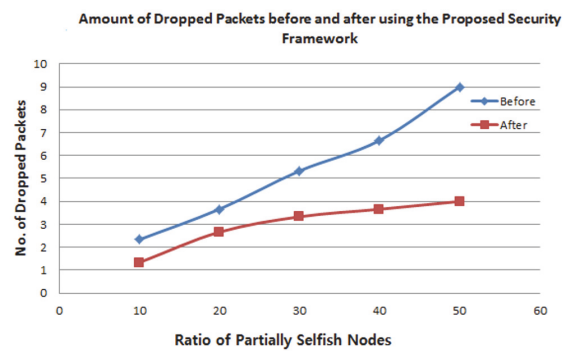

Figure 13: Number of dropped packets while using partially selfish nodes.

was controlled for the simulation. The graphs are averages of 10 runs. In Figures 12 and 13, we can see that, if we increase the number of the selfish nodes in the network, then the number of dropped packets increases simultaneously. We can see that without the proposed solution 27 packets were dropped in the presence of 50 % full selfish nodes in the network, while it was reduced to only 12 after utilizing the proposed solution.

Figure 13 shows reduction in dropped packets by our proposed security framework after considering partially selfish nodes instead of fully selfish nodes. For example, when 50 % of entire nodes start partially selfishly behaved, then we can see that without the proposed solution 9 packets were dropped in the presence of 50 % partially selfish nodes in the network, while it was reduced to only 4 after utilizing the proposed solution. In both cases, we can see the reduction

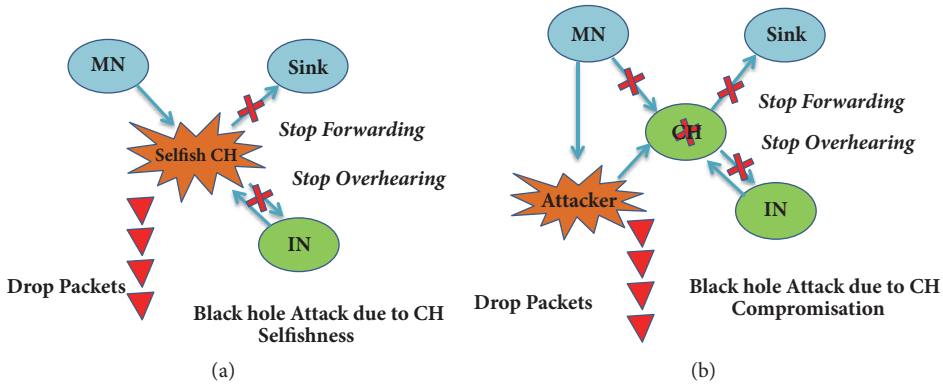

Figure 14: Black hole attack.

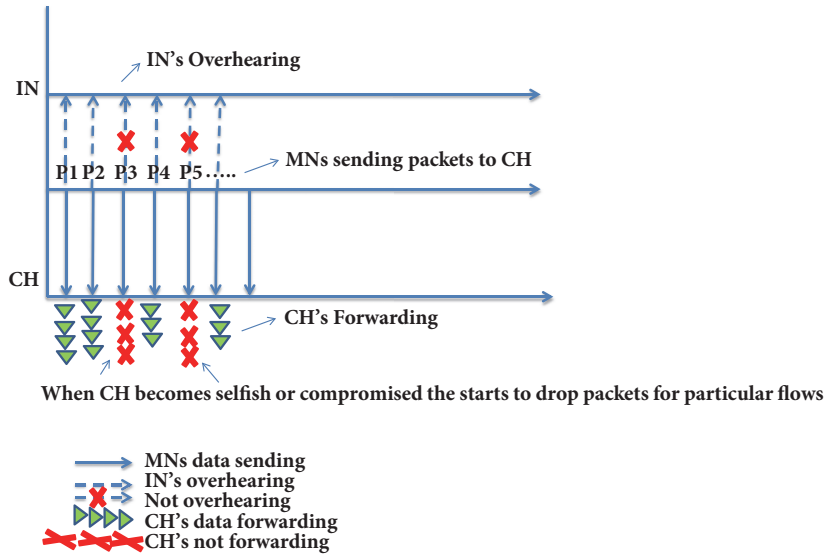

Figure 15: Selective-forwarding attack.

in number of packet drop after using the proposed security framework.

5. Discussion on Active Attacks and Overhead

As a side benefit, the proposed method is also effective against some kinds of active attacks, including the black hole attack, the on/off attack, the selective-forwarding attack, and the transmission opportunity-wasting attack. (i) According to [23] the black hole attack is a type of Denial of Service (DoS) attacks and attacker easily launch it by capturing and reprogramming a set of nodes in the network. As a result, any information that enters the black hole region is captured and blocked from forwarding to the base station, such that important event information does not reach the base stations and the network performance is degraded. Figure 14 shows the black hole attack scenario in light of our proposed security framework. We can easily inspect that in the proposed cluster the selfish or compromised CH leads to the black hole attack scenario. When MNs send packets to this compromised or selfish CH, it starts to drop them instead of forwarding them to the base station. We claim that this type of situation can be easily monitored and controlled by overhearing by IN. (ii) The on/off attack means that malicious entities behave well and badly alternatively, hoping that they can remain undetected while causing damage and attempt to disturb a trust redemption scheme [24]. It means that while badly behaving they can act as black holes and start to drop packets instead of forwarding them to the base station. Thus, we urge that it can also controllable by the IN overhearing phenomena. (iii) The selective-forwarding attack keeps a relatively low profile compared with the black hole attack. It drops packets routed to them only for particular flows [25]. It also means that when some nodes to get compromised, they start behaving like the black hole attack. This situation can be seen in Figure 15, where the compromised CH drops packets for the particular flow, say P3. We also argue that this situation can be detected by analyzing the data overheard by IN. (iv) The transmission opportunity-wasting attack simply abandons its scheduled transmission opportunity to degrade network throughput [25]. We can see the same situation in our proposed scheme, where CH becomes selfish or get

compromised, starting to behave like this attack. Depending on the nature of the attack the CH drops packets in different manners. We have observed that these types of situations can be easily detected and controlled by IN overhearing the transmission of CH. Thus, we can say that proposed security framework cannot only control the selfishness attack but also prevent these attacks.

So far, we have observed that the proposed scheme works correctly and prevents the selfishness effectively, resulting in reduce packets drop. However, we also know that the proposed scheme requires additional message overhead as compared to the clustering architecture without the selfishness prevention. Messages are generated largely in the following three cases: (i) CH election, (ii) IN nomination and selection, and (iii) accusation on CH by IN. First, CH election is performed periodically, so the overhead is the same as other clustering networks where CHs are newly elected in every period. On the other hand, for the other types of messages, the overhead is proportional to the ratio of selfish nodes to the total number of nodes in the network. If a selfish node takes the role of IN and it is discovered by CH, then a new IN should be selected. Or, if a CH is selfish, an accusation message should be sent by IN to MNs within its communication range. Thus, the message overhead of the proposed method totally depends on the rate of selfish nodes in the network.

6. Conclusions

We proposed a new security framework against the selfishness attack for CBWSNs. The specialty of this scheme comes from with the appointment of two special nodes (CH and IN) per cluster and the addition of a reputation scheme to every node. CH and IN monitor each other, and MNs watch CH and IN. Through the simulation, we observed that the proposed method can improve the efficiency of CBWSNs not only by controlling the selfishness attack but also by constructing more consistent and resilient clusters. Furthermore, it has a side benefit, restraining on the black hole attack, the selective-forwarding attack, the on/off attack, and the transmission opportunity-wasting attack. In the future, we will extend the idea to the Internet of Things (IoT) domain where smart objects independently run data mining algorithms to analyze other node behaviors and identify the selfishness attack.

Data Availability

The specific data set was not used and not needed actually. The performance study was based on the data randomly generated by the simulation code.

Disclosure

Parts of this study have been presented in ICUFN 2015.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by Mid-Career Researcher Program through NRF grant funded by the MEST (2016R1A2B4016588).

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.

[3] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: recent developments and deployment challenges," *IEEE Network*, vol. 20, no. 3, pp. 20–25, 2006.

[4] P. Kumarawadu, D. J. Dechene, M. Luccini, and A. Sauer, "Algorithms for node clustering in wireless sensor networks: a survey," in *Proceedings of the 4th International Conference on Information and Automation for Sustainability (ICIAFS '08)*, pp. 295–300, Colombo, Sri Lanka, December 2008.

[5] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[6] P. Schaffera, K. Farkas, Á. Horváth, T. Holczer, and L. Buttyán, "Secure and reliable clustering in wireless sensor networks: a critical survey," *Computer Networks*, vol. 56, no. 11, pp. 2726–2741, 2012.

[7] Q. Dong and D. Liu, "Resilient cluster leader election for wireless sensor networks," in *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2009*, Italy, June 2009.

[8] T. Thein, S.-D. Chi, and S. P. Jong, "Increasing availability and survivability of cluster head in WSN," in *Proceedings of the 3rd International Conference on Grid and Pervasive Computing Symposia/Workshops, GPC 2008*, pp. 281–285, China, May 2008.

[9] M. Yan, L. Xiao, L. Du, and L. Huang, "On selfish behavior in wireless sensor networks: a game theoretic case study," in *Proceedings of the 3rd International Conference on Measuring Technology and Mechatronics Automation (ICMTMA '11)*, pp. 752–756, Shangshai, China, January 2011.

[10] Y. Yoo and D. P. Agrawal, "Why does it pay to be selfish in a MANET?" *IEEE Wireless Communications Magazine*, vol. 13, no. 6, pp. 87–97, 2006.

[11] Y. Yoo, S. Ahn, and D. P. Agrawal, "Impact of a simple load balancing approach and an incentive-based scheme on MANET performance," *Journal of Parallel and Distributed Computing*, vol. 70, no. 2, pp. 71–83, 2010.

[12] X. Li, F. Zhou, and J. Du, "LDTS: a lightweight and dependable trust system for clustered wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 924–935, 2013.

[13] H. Yu, Z. Shen, C. Miao, and D. Niyato, "A survey of trust and reputation management systems in wireless communications," *Proceedings of the IEEE*, vol. 98, no. 10, pp. 1755–1772, 2010.

[14] H. Miranda and L. Rodrigues, "Friends and foes: Preventing selfishness in open mobile ad hoc networks," in *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops, ICDCSW 2003*, pp. 440–445, USA, May 2003.

[15] P. Resnick and R. Zeckhauser, "Trust among strangers in internet transactions: empirical analysis of eBay's reputation system," *Advances in Applied Microeconomics*, vol. 11, pp. 127–157, 2002.

[16] K.-F. Ssu, C.-H. Chou, and L.-W. Cheng, "Using overhearing technique to detect malicious packet-modifying attacks in wireless sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2342–2352, 2007.

[17] J. Paek, K. Chintalapudi, R. Govindan, J. Caffrey, and S. Masri, "A wireless sensor network for structural health monitoring: Performance and experience," in *Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors, EmNetS-II*, pp. 1–10, Australia, May 2005.

[18] Z. Ishaq, S. Park, and Y. Yoo, "A security framework for Cluster-based Wireless Sensor Networks against the selfishness problem," in *Proceedings of the 7th International Conference on Ubiquitous and Future Networks, ICUFN 2015*, pp. 7–12, Japan, July 2015.

[19] A. R. Chowdhury, T. Chatterjee, and S. DasBit, "LOCHA: A Light-weight One-way Cryptographic Hash Algorithm for Wireless Sensor Network," *Procedia Computer Science*, vol. 32, pp. 497–504, 2014.

[20] L. B. Oliveira, A. Ferreira, M. A. Vilaça et al., "SecLEACH-on the security of clustered sensor networks," *Signal Processing*, vol. 87, no. 12, pp. 2882–2895, 2007.

[21] M. Sirivianos, D. Westhoff, F. Armknecht, and J. Girao, "Non-manipulable aggregator node election protocols for wireless sensor networks," in *Proceedings of the 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, WiOpt 2007*, Cyprus, April 2007.

[22] S. Vasudevan, B. DeCleene, N. Immerman, J. Kurose, and D. Towsley, "Leader election algorithms for wireless ad hoc networks," in *Proceedings of the DARPA Information Survivability Conference and Exposition, DISCEX 2003*, pp. 261–272, USA, April 2003.

[23] G. Gulhane and N. Mahajan, "Performance evaluation of wireless sensor network under black hole attack," *Inter- national Journal of Computing and Technology*, pp. 92–96, 2014.

[24] Y. Chae, L. C. DiPippo, and Y. L. Sun, "Trust management for defending on-off attacks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 1178–1191, 2015.

[25] Z. Lu, Y. E. Sagduyu, and J. H. Li, "Queuing the trust: Secure backpressure algorithm against insider threats in wireless networks," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks, IEEE INFOCOM 2015*, pp. 253–261, Hong Kong, May 2015.