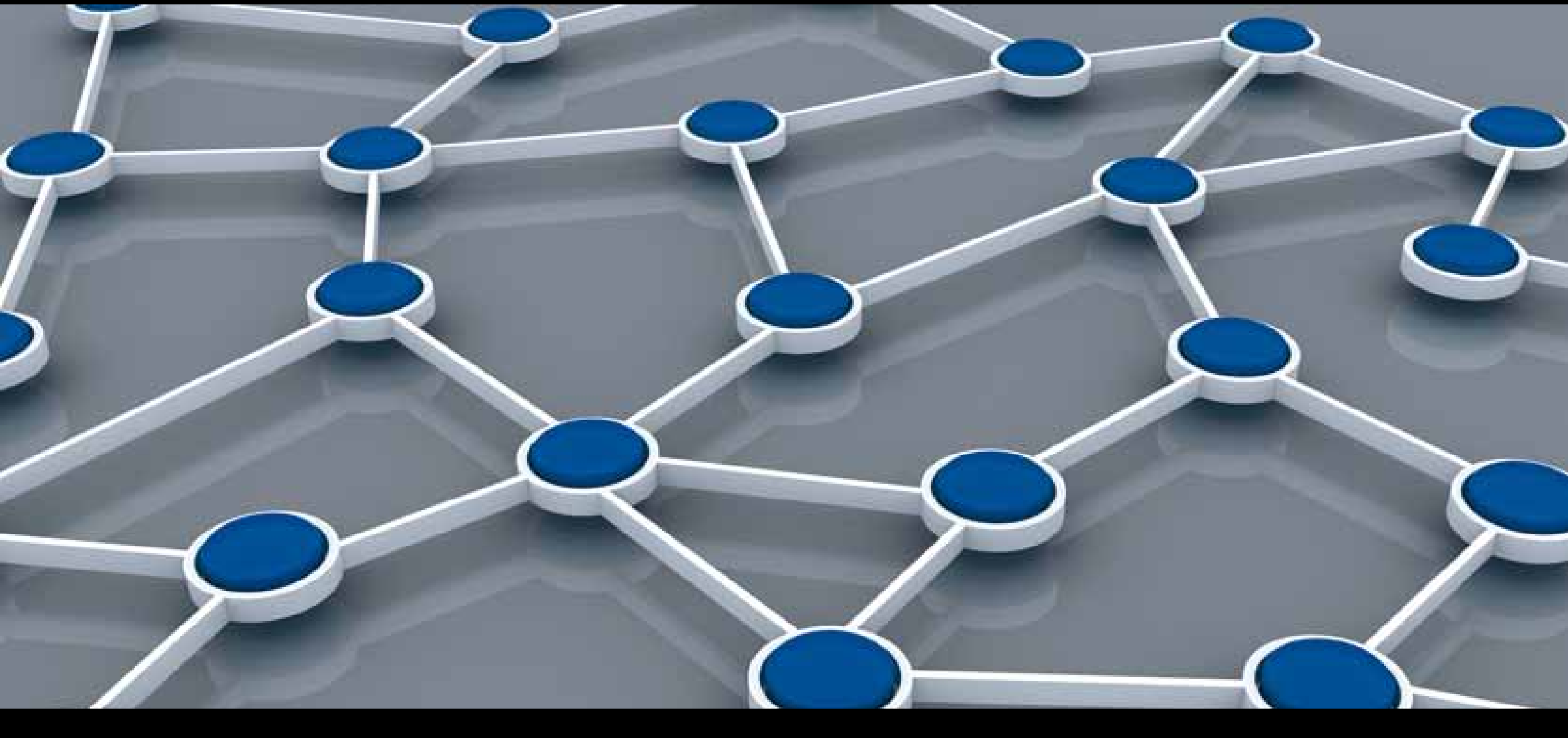
International Journal of Distributed Sensor Networks

Intrusion Detection and Security Mechanisms for Wireless Sensor Networks

Guest Editors: S. Khan, Jaime Lloret, and Jonathan Loo



Intrusion Detection and Security Mechanisms for Wireless Sensor Networks

International Journal of Distributed Sensor Networks

Intrusion Detection and Security Mechanisms for Wireless Sensor Networks

Guest Editors: S. Khan, Jaime Lloret, and Jonathan Loo

Editorial Board

Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 747483, 3 pages
http://dx.doi.org/10.1155/2014/747483

Editorial

Intrusion Detection and Security Mechanisms for Wireless Sensor Networks

S. Khan,[1] Jaime Lloret,[2] and Jonathan Loo[3]

[1] *Institute of Information Technology, Kohat University of Science and Technology (KUST), Kohat City 26000, Pakistan*
[2] *Integrated Management Coastal Research Institute, Universidad Politecnica de Valencia, Camino de Vera, 46022 Valencia, Spain*
[3] *School of Science and Technology, Middlesex University, The Burroughs, London NW4 4BT, UK*

Correspondence should be addressed to S. Khan; skhan@kust.edu.pk

Received 2 February 2014; Accepted 2 February 2014; Published 4 March 2014

Wireless sensor networks are multihop, self-organizing, self-healing, and distributed in nature. One of their main features is their energy consumptions, so many efforts are focused on power saving techniques. Wireless sensor networks are gaining significant interest from academia and industry and the number of real deployments of wireless sensor networks (WSN) is increasing considerably in the last years. Their intrinsic characteristics make them very vulnerable to external intrusion. Thus, the security has become one of the main issues to study in WSNs. Their ad hoc network nature also increases their vulnerability and exposes sensor nodes to various kinds of security attacks. There is a clear need for new security techniques to guarantee the information transmitted through the WSN. Last research tendencies are focused on including security in the routing protocol, providing security for communication inside groups of nodes and when exchanging data between groups. One of the most efficient techniques to detect an intruder in the network is the use of traffic analysis for detecting anomalies and finding correlated events. Advanced security mechanisms and intrusion detection systems (IDSs) can play an important role in detecting and preventing security attacks in WSNs.

In this special issue, we have collected recent advances in intrusion detection and security mechanisms for WSNs. The papers have been peer reviewed and have been selected on the basis of their quality and relevance to the topic of this special issue.

There are many WSN applications; however, such networks are highly vulnerable to different security attacks such as altering, misusing, or misdirecting the data in transit. Keep these issues in mind. M. Usama and F. T. B. Muhaya present the paper "*Framework for secure wireless communication in wireless sensor networks*." The framework consists of few modules such as redundancy checker, message prioritization mechanism, malicious node verification, and malicious data verification. It is evaluated and validated using NS2 simulator. The experimental results show that the proposed secure framework can be used for malicious data or node detection.

There are many techniques which are used to design IDSs for WSNs. Artificial intelligence techniques are widely used for this purpose. In the paper "*Intrusion detection systems based on artificial intelligence techniques in wireless sensor networks*," a critical study on genetic algorithm, artificial immune, and artificial neural network (ANN) based IDSs techniques used in wireless sensor network (WSN) is presented. ANN concepts are helpful in many areas such as pattern recognition and intrusion detection. ANN based intrusion detection can be helpful to eliminate the shortcomings of rule based IDSs. However, ANN based IDSs can be more effective if properly trained with both normal and abnormal data sets.

In the paper titled "*Lightweight and scalable intrusion trace classification using interelement dependency models suitable for wireless sensor network environment*," a lightweight and scalable method for classification of network and detection of system intrusion attempts is presented. The authors claim better intrusion detection accuracy of the proposed system. The mechanism involves many actions such as

Naive Bayes Classifier, Naive Bayes with n-gram features, interdependency modeling, Support Vector Machines, and k-truncated suffix tree. Experiments are conducted with many data sets such as UNM and MIT LL 1998.

The authors of the paper *"Intrusion detection systems in wireless sensor networks: a review"* present a review of IDSs in WSN such as the following.

(i) *Signature based IDSs*—signature based IDSs are also known as rule based as they have predefined rules for different security attacks. These IDSs can only detect those security attacks whose signatures are present in the databases. The limitation of such detection mechanisms is that they cannot detect new or those attacks whose signatures are not present.

(ii) *Anomaly based IDSs*—anomaly based IDSs are intelligent and they do not have a support of predefined rules. These kinds of IDSs mostly use threshold value for classifying attacks.

(iii) *Hybrid IDSs*—hybrid IDSs use both mechanisms; that is, it is the combination of signature based approach and anomaly based approach. However, hybrids IDSs are not considered suitable for WSNs due to more complexity and heavyweight nature.

(iv) *Cross-layer IDSs*—as multihop wireless networks are vulnerable to multi-layer security attacks, and security mechanism for one layer cannot detect and respond to a security attack at another layer. Cross-layer IDSs are capable of detecting multilayer security attacks. However, as WSNs are resource constraint networks, that is why cross-layer IDSs are not considered suitable for such networks. The reason is that cross-layer IDSs exchange parameters across the protocol stake which may consume more power, memory, and processing.

Key management in a large cluster based sensor network with limited resources is a challenging task. Furthermore, an attacker can compromise the entire network security just controlling few nodes and injecting false data in an undetected manner. To protect WSNs from such impersonating attacks, G. Jeong et al. present a new dynamic key management framework particularly for large-scale clustered sensor networks in the paper "*Impersonating-resilient dynamic key management for large-scale wireless sensor networks*." In the framework, different keying mechanisms, respectively, secure in-cluster, intercluster, and individual communication by refreshing keys on demand, while adaptively handling node addition and capture. Experimental results show that the proposed framework provides higher security against impersonating attacks with less cost. The scheme can efficiently handle many scenarios such as node addition and eviction. The proposed solution also presents different keying mechanisms for different situations such as in-cluster, intercluster, and individual communication. It uses simple cluster-shared one-way key chain.

An eavesdropping attack with directional antennas in WSN is described in the paper "*On eavesdropping attacks in wireless sensor networks with directional antennas*." The proposed model analyzes the eavesdropping probability in both single-hop WSNs and multihop WSNs with omnidirectional antennas and directional antennas. It is demonstrated that to use directional antennas can significantly reduce the eavesdropping probability in both single-hop and multihop WSNs. Directional antennas provide more security due to the smaller region and the low number of hops in the route. The experimental results show that better security is dependent on the signal path loss factor, beam width, and node density. The paper has some interesting contributions such as

(i) eavesdropping modeling in WSNs with omnidirectional and directional antennas,

(ii) determination of exposure region to check whether an attacker can eavesdrop or not,

(iii) analysis of eavesdropping attacks in both single-hop networks and multihop networks with both omnidirectional antennas and directional antennas.

A game theoretic model for hidden-action attacks is proposed in the paper "*A game theoretic model for wireless sensor networks with hidden-action attacks*." This mechanism investigates the attack and detection problem by modeling it as pairwise simultaneous game and spatial structured game. In experimental results, different effects are analyzed such as

(i) action cost effect,

(ii) coaction effect,

(iii) stimulation effect,

(iv) punishment effect.

The authors are interested in extending this work in the future by adding some important features such as

(i) further enhancing the efficiency of the proposed mechanism to detect many other attacks,

(ii) estimating the cost of the game modeling in WSN,

(iii) considering more complex WSN models.

Security in cognitive wireless sensor networks (CWSN) is an important problem because these kinds of networks are used in critical and important applications. Moreover, the limited resources of WSN make the problem even more complex. The paper "*PUE attack detection in CWSN using collaboration and learning behavior*" is focused on primary user emulation (PUE) attack in CWSN. In PUE attack, a malicious node emulates the behavior of an incumbent node to use the radio spectrum for its own useless operation or stop the other nodes to access the spectrum. The authors claim that in most of the previously published works use traffic monitoring to train a behavior model of the network, while, in this work, other parameters such as power transmission to detect anomalies in CWSN are also considered. Many experiments are conducted to validate the behavior of the proposed system; it is observed that if the collaborative nodes are over 20% of the total, the PUE attack detection has satisfactory results, with 98% of attacks detected and a false negative rate near 0%.

The paper "*Evaluation, energy optimization, and spectrum analysis of an artificial noise technique to improve CWSN security*" focuses on physical layer security in CWSN. It discusses many aspects such as energy optimization, security evaluation, and spectrum analysis of artificial noise techniques to strengthen the physical layer security. The authors claim that these techniques introduce noise into the spectrum to hide real information. The important finding of this paper is an intelligent mechanism, in which the transmitter with the help of few supporting nodes is capable of generating noise in order to hide the data in transit. The noise is generated in such a way that only the attacker is affected, not the legitimate receiver.

External attacks can be detected easily as compared to internal attacks. In internal attack(s), the attacker is inside and most of the time is a legitimate member of the network. Detection of insider malicious node is a challenging task as it not only is a legitimate member of the network but also knows exactly what their neighbors or monitoring nodes know. Such insider malicious node can launch attacks secretly and carefully to avoid being detected and discarded from the network. In the paper titled "*Detection and prevention of selective forwarding-based denial-of-service attacks in WSNs*," a mechanism to detect insider malicious node capable of selective forwarding-based denial-of-service attack is proposed. The proposed scheme uses trust management approach to detect such malicious nodes and their victims. Furthermore, the authors designed two attacker-aware protocols to route victim nodes packets using alternate path to avoid attackers. The paper presents some interesting findings.

(i) Beta trust model and entropy trust model fail to detect selective forwarding attack.

(ii) The potential damage to the network due to such attacks is analyzed.

(iii) A source level trust management scheme is used to enhance the security features of beta and entropy models.

(iv) An avoidance strategy is presented to reroute the packets so as to avoid attackers.

(v) A preventive mechanism is also presented to prevent such attacks.

Adaptive ant based secure routing protocol to select two optimal paths keeping in view route security is proposed in the paper "*Secure ant-based routing protocol for wireless sensor network*." The proposed scheme has four important steps, that is, route discovery, route selection, route security, and data forwarding to destination. This mechanism is inspired from the real ants. Forward and backward ants are used for route request and route reply purpose. Furthermore, the forward ants collect and increment the reputation values along the path to ensure security. The scheme is compared with other candidate solutions such as LEACH and iACO protocols.

In the paper "*On lightweight intrusion detection: modeling and detecting intrusions dedicated to OLSR protocol*," a signature based IDS in cooperation with OLSR routing protocol is presented. Most of the existing IDSs monitor the ongoing traffic for attacks detection. Contrary to other solutions, the proposed system monitors and analyses logs for misuse detection.

S. Khan
Jaime Lloret
Jonathan Loo

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 585491, 8 pages
http://dx.doi.org/10.1155/2013/585491

Research Article

Framework for Secure Wireless Communication in Wireless Sensor Networks

Muhammad Usama[1] and Fahad T. Bin Muhaya[1,2]

[1] *Prince Muqrin Chair for IT Security, King Saud University, Riyadh, Saudi Arabia*
[2] *College of Applied Studies and Community Service, King Saud University, P.O. Box 2459, Riyadh, Saudi Arabia*

Correspondence should be addressed to Muhammad Usama; usama.khanzada@hotmail.com

Received 25 March 2013; Revised 11 July 2013; Accepted 13 September 2013

Academic Editor: S. Khan

With large-scale and rapid development in wireless sensor networks (WSNs), there is great demand to adopt security mechanisms for secure wireless communication. WSNs have many fields of applications that are playing an essential role in increasing productivity and reducing cost. The restricted and constrained nature of sensors along with potentially dynamic behavior of WSNs demands the proper implementation of framework for secure communication in order to prevent attacker from illegally accessing or altering the transmission. We proposed a framework for secure wireless communication in WSNs which consists of four modules, that is, redundancy checker, message prioritization mechanism, malicious node verification, and malicious data verification. Detailed results of security and performance analysis have been realized for comparison and evaluation with complete implementation using NS2 network simulator. The proposed framework presents numerous interesting features which proofs acceptable performance for malicious node and data detection in WSNs.

1. Introduction

Recent advances in wireless senor networks (WSNs), which are typically composed of low power, small microprocessors, powerful base stations, and a large number of multifunctional resource-constrained sensor nodes [1, 2], have attracted significant attention from industries and academics and are an interesting topic of continuous research. These resource-constrained devices provide sensing functionality when they are networked together over a wireless medium. The dynamic, self-directed and distributed network environment of sensor devices in WSNs leads them to depend on sensors functionality. In WSNs, all sensor nodes communicate and collaborate for a special purpose and common goal. They can be used to collect and process certain physical information and data from the environment such as mechanical, temperature, light, radiation, and optical readings, to transmit on base stations. These features and characteristics of WSNs enables wide range of applicability in various applications of environment monitoring, logistic and health care applications, military operations, and so forth [3–7]. WSNs can be configured as heterogeneous, homogeneous, hierarchical, or distributed. It depends upon requirements with available hardware and network settings options. For example, the design of hierarchical wireless sensor network depends on the capabilities and functionalities of base station, head, and sensor devices as shown in Figure 1. Sensor devices are mostly used for special purposes and they require less power and processing time. They collect data from deployed environment and transmit to nearest head device. The head device is responsible for collecting and processing data gathered from near sensors and forwarding it to base station. Base station further processes the data gathered from head devices to make it meaningful for other networks. The base stations are more powerful and can process large data. In distributed or any other wireless sensor network, the capabilities and functionality of all sensors and devices are similar as shown in Figure 2.

In many WSNs applications, these systems are deployed in unattended, self-directed, dynamic and hostile environments. In such systems, message broadcast technique is an efficient and common communication practice to multiple users. Similarly, user can join the host network by queries or

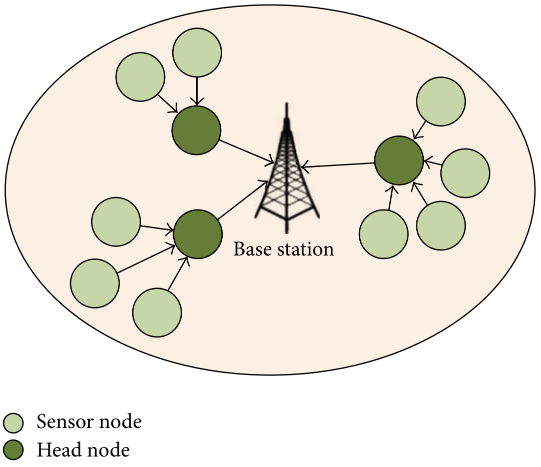


Figure 1: Hierarchical WSNs.

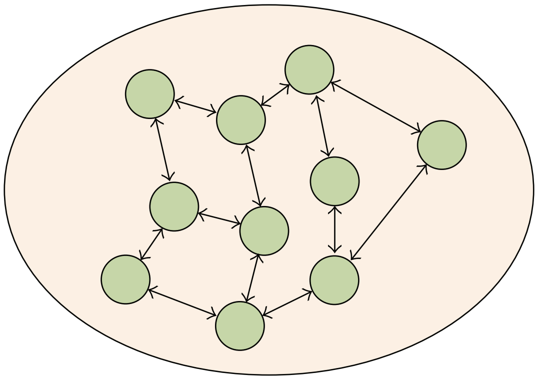


Figure 2: Distributed WSNs.

commands messages for obtaining desired information and data [8, 9]. The use of wireless technology for messages and data transmission requires proper implementation of secure wireless communication framework. Due to the deployment nature and wireless communication, attacker can easily access the transmission, can threaten other users, capture data, and alter or misuse communication illegally. Therefore security becomes serious issue and concern to prevent illegal use or access and protect broadcast messages from various malicious attacks.

In this paper we propose a framework for secure wireless communication to ensure that sensor node communicates secure information to its neighbor through WSN. We evaluate two study scenarios for the broadcast message on WSN to demonstrate the security and performance. The detailed quantitative analysis and experiment show that the proposed framework is greatly superior to the traditional methods and techniques for WSNs in terms of energy consumption and transmission delay of the whole network. It reduces network load using redundancy checker module by removing message redundancy. It is suitable and practical for malicious nodes and detection as compared to existing approaches. Proposed framework supports message prioritization mechanism to ensure more access for network and user traffic as required.

2. Related Work

Many techniques and approaches regarding detection and management of malicious nodes, malicious data, errors, or faults have been published during the last years for secure communication in wireless sensor networks. However, WSNs are susceptible to several security threats and vulnerable to many attacks due to the broadcast nature of data communication. In addition, sensor nodes are more susceptible to attack or danger as they are placed in unattended, self-directed, dynamic, and hostile environments [10]. In [11] Raya and Hubaux presented a security architecture for security and privacy. In [12] Golle et al. proposed a malicious data detection and correction technique based on sensor data and neighbor information. It allows malicious node to perform malicious activities that provides redundant and position information of neighbors in order to detect and correct malicious data. In order to remove malicious nodes, Xiao et al. [13] proposed a scheme based on signal strength that can verify the position to localize and detect. The proposed scheme does not provide acceptable security and comes up with weakness especially against spoof attacks. Again they proposed two static algorithms based on traffic patterns and base stations to overcome the weakness and reduce the effect of malicious nodes.

It is important that system must be able to detect malicious nodes from WSNs with acceptable performance. However, it is difficult to detect malicious nodes due to dynamic nature and lack of proper implementation of WSNs. Many applications suffer because of various security attacks and vulnerabilities, for example, denial of service, malicious node attack, impersonation, copyright and privacy violation, and so forth, due to lack of proper implementation and dynamic nature [14]. Authentication, confidentiality, integrity, and nonrepudiation are also very important and essential security requirements of wireless communication [15]. In denial of service (DoS) attack, attacker attempts to disrupt, subvert, or destroy a network [16]. A DoS attack decrease the network capacity in terms of size, extent, and range in order to reduce network performance from expected functions. Due to the broadcast nature of communication, any attacker can access or use communicated messages and data illegally. Using intrusion or any transmission access techniques, attacker can get important information like location, IDs, timestamps or any specific information, and so forth. WSNs basically adopt neighbor trust model in which sensor nodes rely on neighbor sensor node for message forwarding. In such trust model attacker can exploit communication by selective forwarding attack [17]. Attacker can create malicious node by simply dropping received messages in order to confuse the network transmission that some or all messages are not delivered or dropped. Thus, neighbor should find alternative route for communication, and in the same manners messages can be tampered before forwarding by malicious nodes.

3. Proposed Framework

Generally a secure wireless communication framework requires especial and careful design and implementation that deals with resources constrains while ensuring better

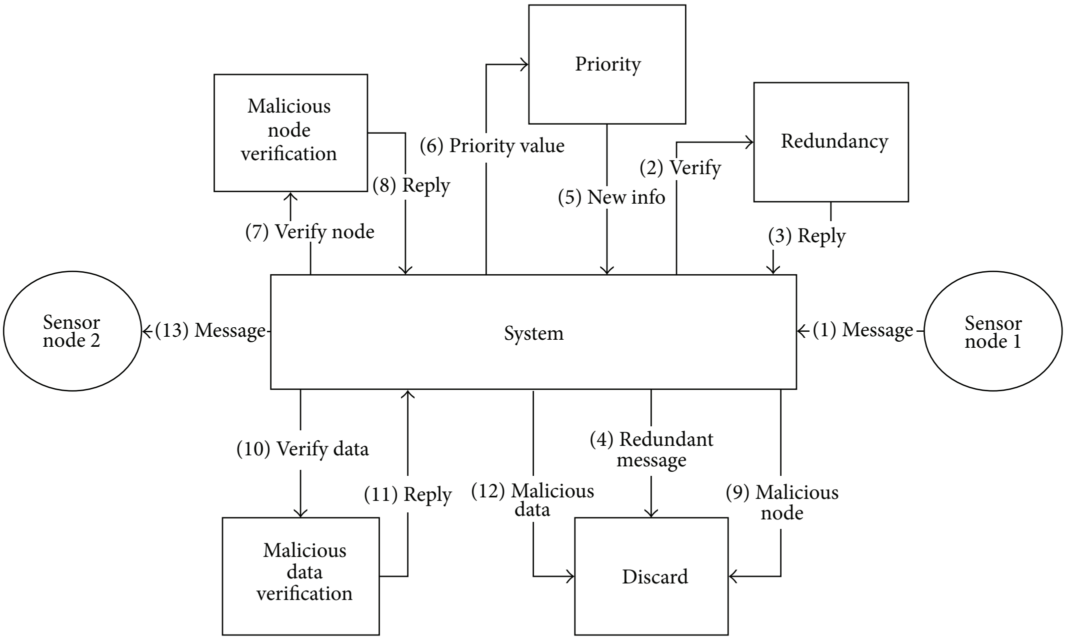


FIGURE 3: System block diagram.

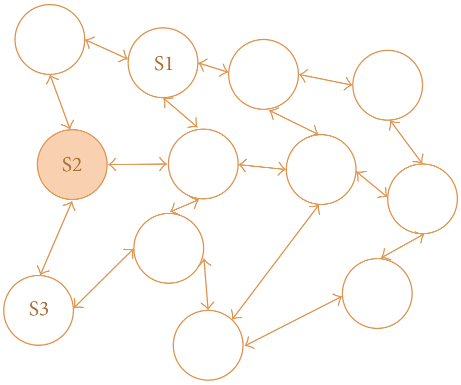


FIGURE 4: Sensors nodes network where S2 is a malicious sensor node.

performance and efficiency along with flexibility and ease to use application access and processes. In order to provide such a secure framework for WSNs we have made a few reasonable assumptions. We assumed that the base stations and head nodes are secure and attackers cannot access or use them illegally. The proposed framework does not aim any sort of trust assumptions on the wireless communication with the apparent fact of nonzero probability of messages deliveries. Secure communication framework design should build robust and trustworthy system with untrustworthy modules and should have ability to perform well when need arises. Security framework should also work with the same features if change occurs in the system of addition or updating in the network nodes, thus providing scalability with minimum errors.

Our proposed secure wireless communication framework for WSNs is composed of four core modules, called redundancy, message prioritization mechanism, malicious node, and data verification. The block diagram of the proposed framework is given in Figure 3. The system performs the following operation steps in sequential manner in order to ensure secure communication.

Step 1. Sensor 1 wants to share some information with Sensor 2 as message.

Step 2. System sends message to redundancy checker for verification and checking of message redundancy.

Step 3. If the message is redundant than system discard else return for forward. At this stage all redundant messages are discarded and only newly issued messages can be forwarded.

Step 4. System forwards new message for assigning priority value. Priority checker returns back priority value to the system.

Step 5. After performing redundancy checking and priorities for the message, it performs malicious node verification for ensuring security. The malicious node verification module replies to forward the message for further verification or discard.

Step 6. If the node is malicious, then data is discarded; otherwise it returns to system.

Step 7. At the end, the system verifies the message for the presence of malicious data and finally decides to forward the message for further communication or discard.

Step 8. If it is found that message contains malicious data then it is discarded; else it is delivered to neighbor node safely.

Table 1: Simulation parameters.

Parameters	Values
Channel	Wireless
Sensor nodes	N
Protocol	MAC 802.11
Radio propagation model	Two-ray ground
Time	50 seconds
Data	Multimedia

The description of basic functions and assumptions of each module is listed below.

(i) Redundancy checker: in this system, each sensor node is responsible for maintaining unique message ID table of received messages. During communication redundancy checker utilizes the unique message ID to detect or discard the redundant messages.

(ii) Message prioritization mechanism: prioritization mechanism computes and compares the message priority values with other messages. By default, it assigns higher priority to all safety messages.

(iii) Malicious node verification: the malicious node verification process is done using signal strength.

(iv) Malicious data verification: the malicious data verification process is done using existing messages and node position.

4. Implementation and Results

This section will briefly provide an overview on the implementation and results of the proposed framework. Simulations have been performed and observed for multimedia streaming in a wireless sensor network scenario. We assume that the sensor nodes are uniformly distributed and deployed in the field with support of an underlying MAC protocol 802.11 and a two-ray ground model for the wireless channel access. A two-ray ground radio propagation model is used that is an empirical mathematical formulation for the characterization of radio wave propagation [18]. It considers direct and ground paths between two wireless sensor nodes using radio waves propagation. Sensor node can access, channel via MAC protocol after exchanging messages with neighbors. The mobility model is implemented using Manhattan Mobility Model [19]. Multimedia traffic is generated using EvalVid [20]. We perform data availability test by measuring time that elapses to send and receive data traffic to ensure data availability for all network nodes. Detailed simulation results of security and performance analysis have been realized for comparison and evaluation with complete implementation using NS-2 [21] on Cygwin [22]. Experiment control parameters with values used in simulation are given in Table 1.

The performance of the proposed framework highly depends on proper implementation and deployment of WSNs. To analyze the performance of the proposed framework, we conduct two studies to verify the secure wireless communication in WSNs and simulate the data traffic and compute throughput, delay, and PSNR for comparison and evaluation.

(i) Throughput: in a random network that consists of N nodes where each node S has a randomly chosen destination node D and can transmit at t bits-per-second, throughput is scaled as

$$\Theta\left(\frac{1}{\sqrt{n\log n}}\right)^{1} \text{ per } S\text{-}D \text{ pair.} \tag{1}$$

A throughput $t > 0$ is feasible if every node in a network can send t bits-per-second.

(ii) Delay: the delay is the time of a packet to reach the destination after leaving source in a network.

(iii) Peak signal noise ratio (PSNR): PSNR is used to measure the error between transmitted multimedia data and the original data. The equation below defines the PSNR between component Y of source S and destination D:

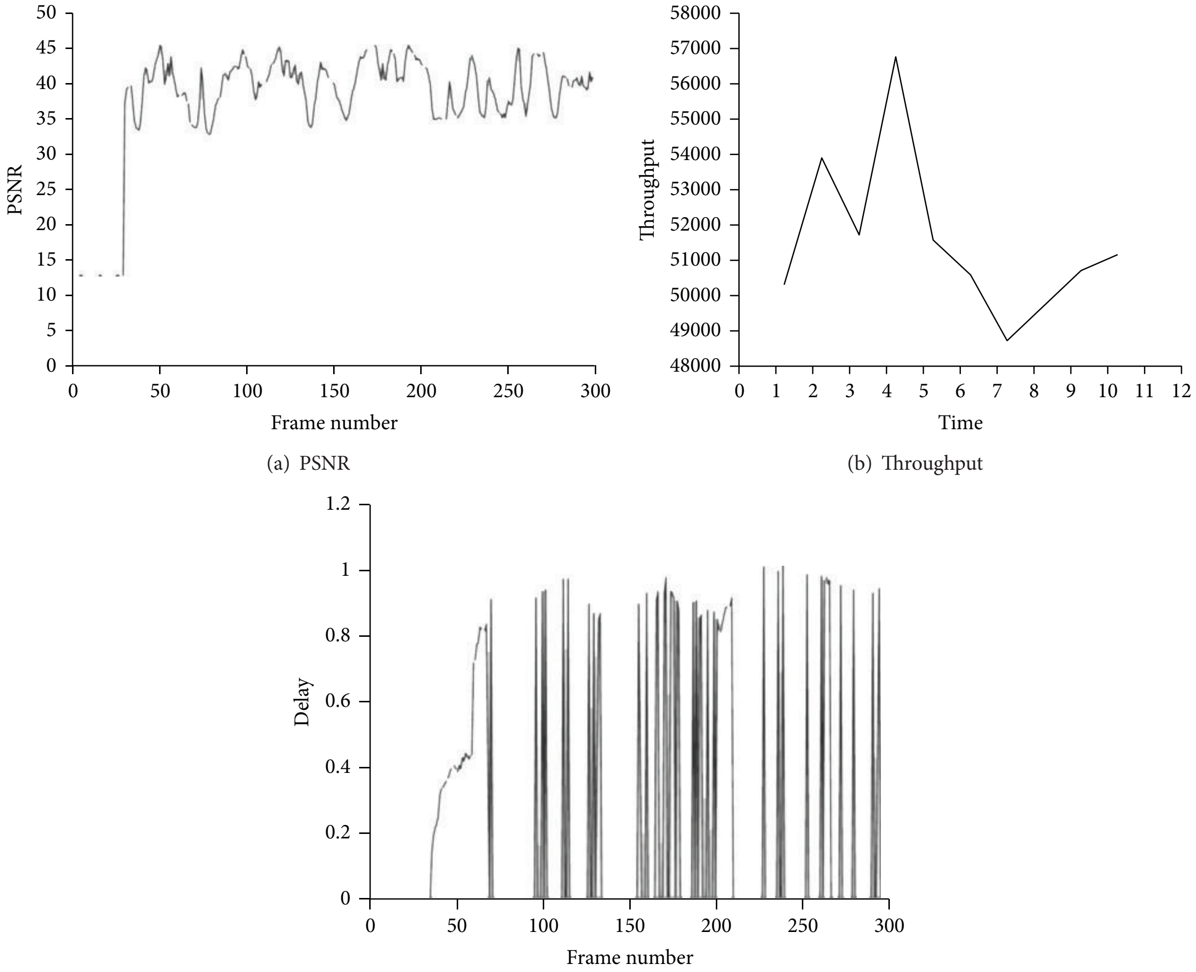
$$\mathrm{PSNR}(n) = 20\log_{10}\left\{\frac{M_{\text{peak}}}{\sqrt{(1/N_{\text{col}}N_{\text{row}})\sum_{i=0}^{N_{\text{col}}}\sum_{j=0}^{N_{\text{row}}}\left[Y_S(n,i,j) - Y_D(n,i,j)\right]^2}}\right\}, \tag{2}$$

where $M_{\text{peak}} = 2^m - 1$ and m: number of bits per pixel.

In the first scenario, we measure the delay, PSNR, and throughput without using the proposed framework. In the second scenario, we perform the same analysis by implementing the proposed framework for detection of malicious node and data. In these simulations, we assume that each sensor can only directly communicate with its neighbor sensors. Sensor node 1 wants to send its secure message to Sensor node 2. The secure message will be communicated through the proposed framework and verified for malicious node and data identification. To give a detailed experiment analysis, we further analyze and compare the throughput, delay, and PSNR in the WSNs.

4.1. Study Scenario I: Performance Analysis in Ideal State.

In this study, we analyze the performance in ideal state (i.e.,



(a) PSNR

(b) Throughput

(c) Delay

Figure 5: Analysis results in ideal state.

no proposed framework to detect malicious data and node), which gives us results for comparative performance analysis when we apply our proposed framework to determine the redundant messages, malicious data, and nodes in WSNs. Here, we have highlighted three sensor nodes: S1, S2, and S3. Sensor nodes S2 and S3 want to share secure data with sensor node S1, where S2 is malicious, that is, sending malicious data to S1 in order to degrade the performance of WSNs as shown in Figure 4. As mentioned, system is not able to validate the malicious activities and data due to lack of secure wireless communication framework. The system assumes that all network nodes are fair. Analysis results are given in Figure 5; the computed delay in this communication is higher as shown in Figure 5(b) and the throughput is lower as shown in Figure 5(c) because S2 is malicious and sends malicious data to S1. Figure 5(a) shows reconstruction peak signal-to-noise ratio (PSNR) for all frames in the network.

4.2. Study Scenario II: Performance Analysis with Proposed Framework. In this study, we simulate the same scenario as we did in study I with the proposed secure wireless communication framework. As mentioned earlier, Sensor nodes S2 and S3 want to share secure data with Sensor node S1, where S2 is a malicious and sends malicious data to S1 in order to degrade the performance of WSNs as shown in Figure 6 but here the message is communicated through secure framework to check message redundancy, determine message priority, and detect and verify malicious node and data.

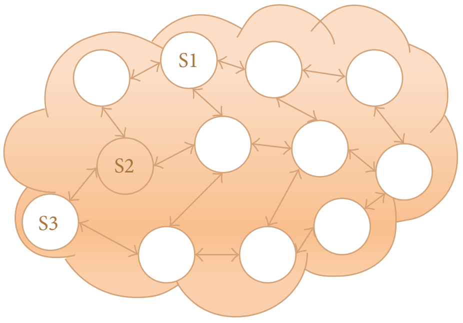
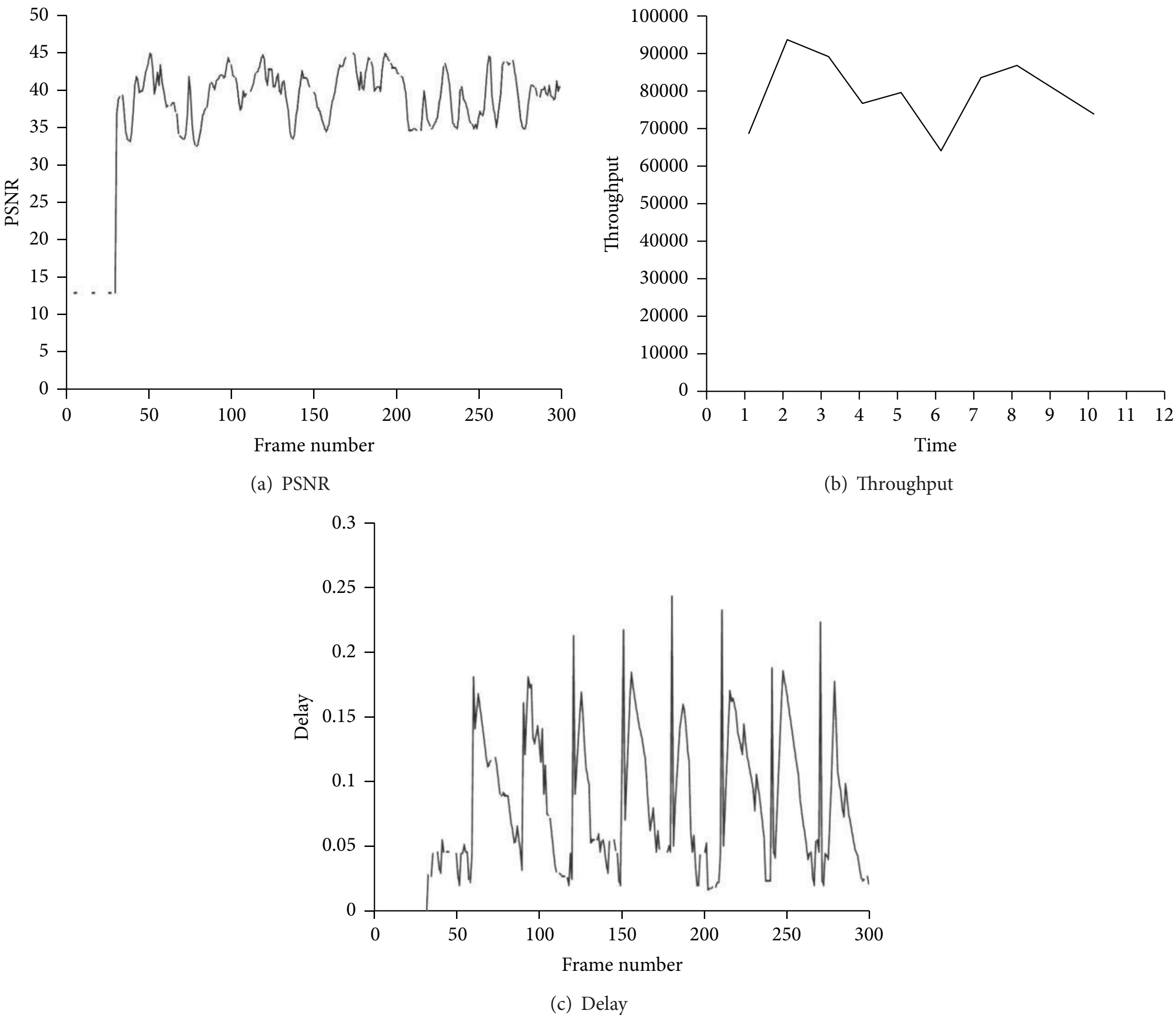


Figure 6: Sensors nodes network with secure wireless communication framework.



(a) PSNR

(b) Throughput

(c) Delay

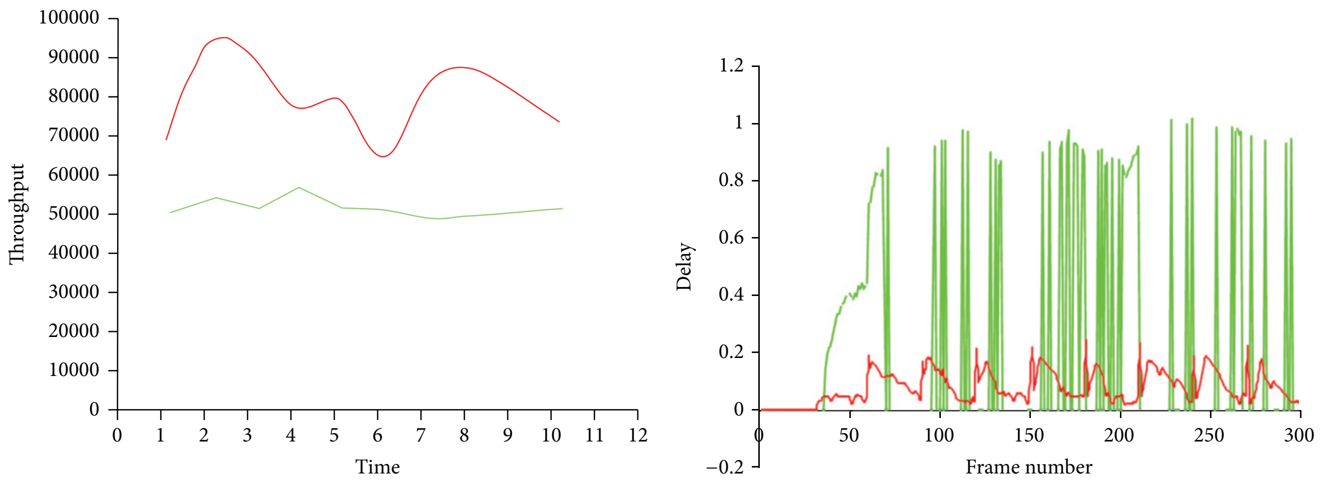
Figure 7: Analysis results with secure wireless communication framework.

The delay, PSNR, and throughput have been computed using the proposed secure wireless communication framework as shown in Figure 7. Experiment results show that sensor network performance is consistent and acceptable. It is stable and suitable to prevent the network degradation with proper implementation. It detects malicious data during message transmission from neighbours. Experiment shows that PSNR analysis results are the same in both simulations as shown in Figures 5(a) and 7(a). Figures 5(b) and 5(c) show that computed throughput is higher and delay is lower, respectively, because framework detects the malicious data without degrading network performance.

4.3. Comparison. In the following, we present and discuss the simulation comparison results. We performed the simulation for both scenariosL: in the first scenario the framework was in ideal state and in the second scenario the framework was present for malicious node and data detection. Simulation parameter values are given in Table 1. In both runs of the simulation, we set Sensor node S2 as a malicious node that is sending malicious data in order to degrade the performance of WSNs. Right before the beginning of malicious activities, all statistics are readjusted to zero. In each scenario, we measure the delay, throughput, and PSNR to determine the level of security and performance of the proposed framework. Comparison graphs of delay and throughput measurements are given in Figure 8. It is clear that the use of the proposed secure wireless communication framework increases the throughput and decreases the delay for the detection of malicious nodes and data. However, it was reversed when we tested the same without the proposed framework. Experiment results indicate that the performance is acceptable for secure wireless communication. Thus, the proposed framework is suitable and practical for real-time use.

5. Conclusions

With large-scale and rapid development in wireless sensor networks (WSNs), there is great demand to adopt security



(a) Throughput

(b) Delay

Figure 8: Comparison results.

mechanisms for secure wireless communication. WSNs have many fields of applications that are playing an essential role in increasing productivity and reducing cost. However malicious attackers can access the transmission, threaten users, and alter or misuse communication illegally due to restricted and constrained nature of sensors with potentially dynamic behavior of WSNs. Therefore security becomes serious issue and concern to prevent illegal use or access and protect communication from various attacks. We proposed a framework for secure wireless communication in WSNs which consists of four modules, that is, redundancy checker, message prioritization mechanism, malicious node verification and malicious data verification. We present and discuss the simulation results of security and performance analysis for comparison and evaluate the proposed framework. Experiment results indicate that the performance of the framework is acceptable for malicious node and data detection in WSNs and suitable for real-time use.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

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

[3] A. Alemdar and M. Ibnkahla, "Wireless sensor networks: applications and challenges," in *Proceedings of the 9th International Symposium on Signal Processing and its Applications (ISSPA '07)*, pp. 1–6, IEEE Computer Society, Washington, DC, USA, February 2007.

[4] T. Arampatzis, J. Lygeros, and S. Manesis, "A survey of applications of wireless sensors and wireless sensor networks," in *Proceedings of the 20th IEEE International Symposium on Intelligent Control—Mediterranean Conference on Control and Automation (ISIC '05)*, pp. 719–724, IEEE Computer Society, Washington, DC, USA, June 2005.

[5] D. Liu and P. Ning, *Security for Wireless Sensor Networks*, Advances in Information Security Series, Springer, 2006.

[6] J. López and J. Zhou, *Wireless Sensor Network Security*, Cryptology and Information Security Series, IOS Press, 2008.

[7] K. Ren and W. Lou, *Communication Security in Wireless Sensor Network*, VDM Verlag Dr. Müller, 2008.

[8] D. Liu, P. Ning, S. Zhu, and S. Jajodia, "Practical broadcast authentication in sensor networks," in *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '05)*, pp. 118–129, July 2005.

[9] K. Ren, W. Lou, K. Zeng, and P. J. Moran, "On broadcast authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 11, pp. 4136–4144, 2007.

[10] A. Hac, *Wireless Sensor Network Designs*, John Wiley & Sons, New York, NY, USA, 2003.

[11] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '05)*, pp. 11–21, Alexandria, VA, USA, November 2005.

[12] P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in VANETs," in *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET '04)*, pp. 29–37, Philadelphia, Pa, USA, October 2004.

[13] B. Xiao, B. Yu, and C. Gao, "Detection and localization of sybil nodes in VANETs," in *Proceedings of the Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS '06)*, Los Angeles, Calif, USA, September 2006.

[14] M. Raya, P. Papadimitratos, and J.-P. Hubaux, "Securing vehicular communications," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 8–15, 2006.

[15] K. Nahrstedt, J. Dittmann, and P. Wohlmacher, "Approaches to multimedia and security," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '00)*, pp. 1275–1278, New York, NY, USA, August 2000.

[16] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[17] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[18] A. Rahim, Z. Shafi Khan, F. T. Bin Muhaya, M. Sher, and T. H. Kim, "Sensor based framework for secure multimedia communication in VANET," *Sensors*, vol. 10, no. 11, pp. 10146–10154, 2010.

[19] F. Bai, N. Sadagopan, and A. Helmy, "The important framework for analyzing the impact of mobility on performance of routing protocols for Ad hoc Networks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 383–403, 2003.

[20] Network Simulator, "NS-2," 2010, http://www.isi.edu/nsnam/ns/.

[21] C.-H. Ke, C.-K. Shieh, W.-S. Hwang, and A. Ziviani, "An evaluation framework for more realistic simulations of MPEG video transmission," *Journal of Information Science and Engineering*, vol. 24, no. 2, pp. 425–440, 2008.

[22] Cygwin, 2010, http://www.cygwin.com/.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 351047, 6 pages
http://dx.doi.org/10.1155/2013/351047

Review Article

Intrusion Detection Systems Based on Artificial Intelligence Techniques in Wireless Sensor Networks

Nabil Ali Alrajeh[1] **and J. Lloret**[2]

[1] *Biomedical Technology Department, College of Applied Medical Sciences, King Saud University, Riyadh 11633, Saudi Arabia*
[2] *Department of Communications, Universidad Politecnica de Valencia, Camino de Vera, 46022 Valencia, Spain*

Correspondence should be addressed to Nabil Ali Alrajeh; nabil@ksu.edu.sa

Received 13 September 2013; Accepted 21 September 2013

Academic Editor: S. Khan

Intrusion detection system (IDS) is regarded as the second line of defense against network anomalies and threats. IDS plays an important role in network security. There are many techniques which are used to design IDSs for specific scenario and applications. Artificial intelligence techniques are widely used for threats detection. This paper presents a critical study on genetic algorithm, artificial immune, and artificial neural network (ANN) based IDSs techniques used in wireless sensor network (WSN).

1. Introduction

Wireless sensor networks (WSNs) are distributed in nature where sensor nodes operate independently without any centralized authority. Furthermore, sensor nodes have many design and functional limitations in terms of storage, processing, and communication. WSN applications have evolved at very fast pace and are consistently growing in many fields of everyday life [1]. The fast growth of WSN applications demands more ubiquitous and service oriented computing. The fast growing applications, distributed nature, and design limitations of WSNs have resulted in the increase of network related vulnerabilities and threats. There is a need of proper security mechanisms for protecting sensor networks against potential security threats and attacks [2, 3].

The majority of security mechanisms designed so far for WSNs are capable of packet encryption or require authentication to restrict malicious user's access. Furthermore, many other security related solutions for WSNs have been proposed such as authentication, key exchange, and secure routing to prevent some specific attack. However, such security measurements cannot provide a wide range of protection against variety of security attacks and threats against WSNs. An IDS is one possible solution to address a wide range of security attacks in WSNs [4]. IDSs are passive in nature; that is, they are capable of detecting intrusion; however, they cannot prevent or defuse security attacks.

Research community is working on developing new methodologies for intrusion detection in WSNs [5]. Artificial intelligence provides many important and comparatively low cost techniques for designing IDSs while taking care of the energy consumption [6]. Few important intelligent techniques are artificial immune system, artificial neural network, and genetic algorithms.

IDSs based on artificial immune systems are inspired from human immune system. Human immune system is a complex defense system which has the capability to protect human body from foreign germs and other microorganisms. ANN based IDSs are inspired from human neuron system. An ANN is composed of number of neurons (basic processing elements) that are interconnected with each other through weighted connections. Genetic algorithm based IDSs utilize biological concepts of natural selection.

This paper presents short review of the existing intrusion detection mechanisms which are based on artificial intelligence techniques such as artificial immune system, artificial neural network, and genetic algorithm.

The paper is organized as follows. Section 2 analyzes IDSs based on artificial immune system. Section 3 shows the artificial neural network based IDSs. Genetic algorithm based

Table 1: Comparison of immune based IDSs.

IDS	Immune base mechanism	Accuracy	Lightweight	Attack types
[8]	DCA	N/A	No	ICP
[9]	DCA	Below 70%	No	ICP
[10]	Coevolutionary	Moderate	Yes	DoS, R2L, U2R
[11]	Adaptive immune	Above 75%	No	Network layer anomalies

IDSs are studied in Section 4. In Section 5, we conclude the paper.

2. Artificial Immune Based IDSs

This section investigates the applications of artificial immune based IDSs in WSNs. Artificial immune technique provides anomaly based detection of security threats against WSN. Human immune system is a complicated security system which safeguards the human body against many invisible organisms. Human immune system is very complex and consists of dendritic cells (D cells), T cells, and B cells [7].

D cells move in blood and collect information about antigens and dead cells. These cells basically activate response system. T cells are manufactured in bone marrow and are used to destroy infectious cells present in blood. B cells are white blood cells which are responsible for production of antibodies. Nowadays, artificial immune based systems have many applications in computing such as systems optimization, data classification, and intrusion detection.

The general architecture of IDS using artificial immune system is present in Figure 1.

Many works have been done in intrusion detection using artificial immune system.

An immune based mechanism for WSN is present in [8]. The authors claim that WSN is an area where immune based mechanisms can be applied easily. It is based on dendritic cell algorithm (DCA). Furthermore a new security threat called interest cache poisoning (ICP) attack is introduced. ICP is a network layer attack which is capable of disrupting routing packets. Both the DCA and directed diffusion are implemented on every sensor node to perform two tasks, that is, detection of misbehaving nodes and detection of antigens. The direct diffusion mechanisms are as follows:

(i) maintain two tables, that is, interest cache and data cache;

(ii) handle two types of packets, that is, interest packets and data packets.

The DCA monitors different types of signals such as danger signals and safe signals to detect ICP attack. This mechanism was evaluated and tested later on [9] to detect ICP attack; however, the detection rate was below 70%.

Another intrusion prediction technique based on coevolutionary immune system is present in [10]. This mechanism combines both coevolutionary immune system and grid computing features. The proposed system is compared and evaluated with pure immune system. The results show that the proposed system has better learning and understanding capabilities as compared to pure immune system. Furthermore, the proposed mechanism has better detection rate of variety of security attacks such DoS, R2L, U2R, and probing.

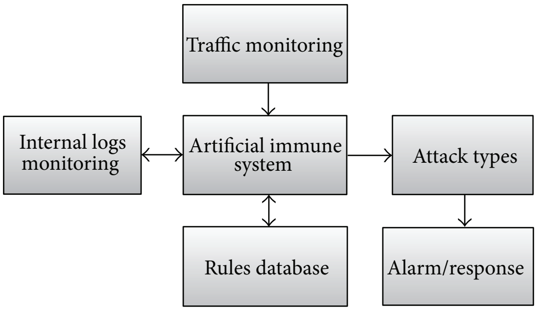


Figure 1: General architecture of IDS using artificial immune system.

Another proposal is present in [11] which is based on adaptive threat detection using immune system. In this work, a MAC layer gene is identified which is helpful in intrusion detection. Every node is equipped with natural selection algorithm to monitor the traffic behavior of surrounding neighbors. This mechanism is capable of detecting network layer anomalies such as packet drop. The detection accuracy of the system is below 85%.

Table 1 presents brief comparison of few immune based IDS.

As we know, WSNs are resource constraints such as memory, processing, and energy. Such constraints demand lightweight IDSs for such networks. In our opinion, the detection mechanisms present in [8, 9, 11] are not suitable for WSNs as they are involved in many processes and computations. The detection mechanism [8, 9] maintains two tables and handles two types of packets. Furthermore different types of signals are monitored for the detection of only ICP attack. Such kind of complex computations for the detection of one attack is not feasible for resource constraint WSNs.

The detection mechanism [11] is involved in identification of MAC layer gene and routing layer security attacks. The exchange of parameters between two different layers needs more resources in terms of memory, computation and energy.

3. Artificial Neural Network Based IDSs

Artificial neural network (ANN) is inspired from human nervous system, which is connected through neurons. Neural networks have the capability to understand and learn by training and can be used to identify complex trends. There

Table 2: Comparison of ANN based IDSs.

IDS	ANN based mechanism	Accuracy	Lightweight	Attack types
[12]	Backpropagation	Above 90% (in many cases)	No	Flooding
[13]	Unsupervised ANN	Above 80%	No	Time related changes
[14]	Feedforward	High	Yes	Malicious nodes

are two types of ANN architectures, that is, feedforward ANN and feedback ANN.

In feedforward ANN, the signals move in only one direction from input to output. In feedback ANN, the signals move in both directions.

ANN concepts are helpful in many areas such as pattern recognition and intrusion detection. ANN based intrusion detection can be helpful to eliminate the shortcomings of rule based IDSs. However, ANN based IDSs can be more effective if properly trained with both normal and abnormal data sets.

The general architecture of artificial neuron is given in Figure 2.

Many intrusion detection mechanisms based on ANN have been proposed for WSNs. In [12], ANN based mechanism for detection of energy exhaustion attack is proposed. The system is for cluster based WSN in which all the sensor nodes are capable of energy harvesting. It consists of three layers, that is, input layer, hidden layer, and output layer. It is trained to discriminate normal and abnormal events. The detection rate for routing attacks is above 95%; however, low detection rate is observed in case of channel access attack.

Intruder detection in WSN with an intelligent mobile robot response system is present in [13]. This mechanism uses an unsupervised neural network for intrusion detection. It detects time related changes using Markov model. Once an intruder is detected, a robot travels to the infected site for investigation. The results show that the detection rate is approximately 85%. As WSN nodes are deployed in harsh and unattended environment where the attacker can manage to compromise few of the nodes, few of the sensor nodes that are compromised may result in wrong data forwarding to the sink. Malicious node detection mechanism using ANN for WSN is proposed in [14]. This mechanism is designed for hierarchical WSN, where nodes sense and share information with neighbor nodes. This mechanism is based on feedforward technique of ANN. The authors claim that the proposed mechanism successfully identified malicious nodes even if 25% of sensor nodes are compromised.

Table 2 presents comparison of few ANN based IDSs.

Any mechanism which is designed for WSN should take care of limited resources of sensor nodes. In our opinion, the detection mechanisms which are present in [12–14] are not suitable for WSNs.

Although the mechanism [12] has high accuracy and the detection rate is above 90%, it has many requirements such as clustering in WSN, energy harvesting, backpropagation algorithm, and training of proposed mechanism.

The mechanism present in [13] is a costly approach as it involves a specialized robot to investigate the infected location.

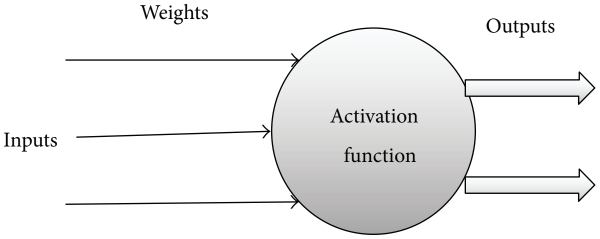


Figure 2: General architecture of artificial neuron.

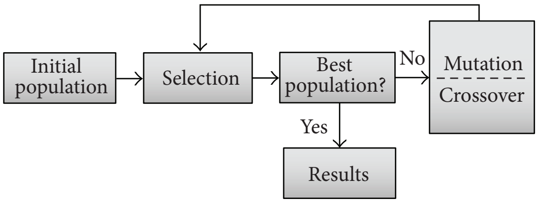


Figure 3: General architecture of genetic algorithm.

The proposal present in [14] seems to be lightweight; however, the claim about the accuracy of the system even in case of 25% malicious nodes needs further investigation.

4. Genetic Algorithm Based IDSs

Genetic algorithms are widely used in many areas of computing to solve a complex problem. It provides robust, adaptive, and optimal solutions for many computing related problems. Genetic algorithms in computing are inspired from biological processes such as natural selection, evolution, theory of mutation, and genetic inheritance.

The general architecture of genetic algorithm used in computing is present in Figure 3.

In genetic algorithm, the selection module derives most suitable answer or solution for some specific problem.

In crossover module, different parameters are exchanged out of different solutions in order to get new solutions.

Mutation module changes one or two parameters to get optimality in genetic algorithm.

Genetic algorithm is widely used technique in network security especially in designing and proposing IDSs. In IDSs, genetic algorithm can be used for classification of security attacks and for generating specific rules for different security attacks [15].

A lightweight IDS with reduced complexity using genetic algorithm for WSN is proposed in [16]. This work deals with measurement of sensor node suitability and attributes to

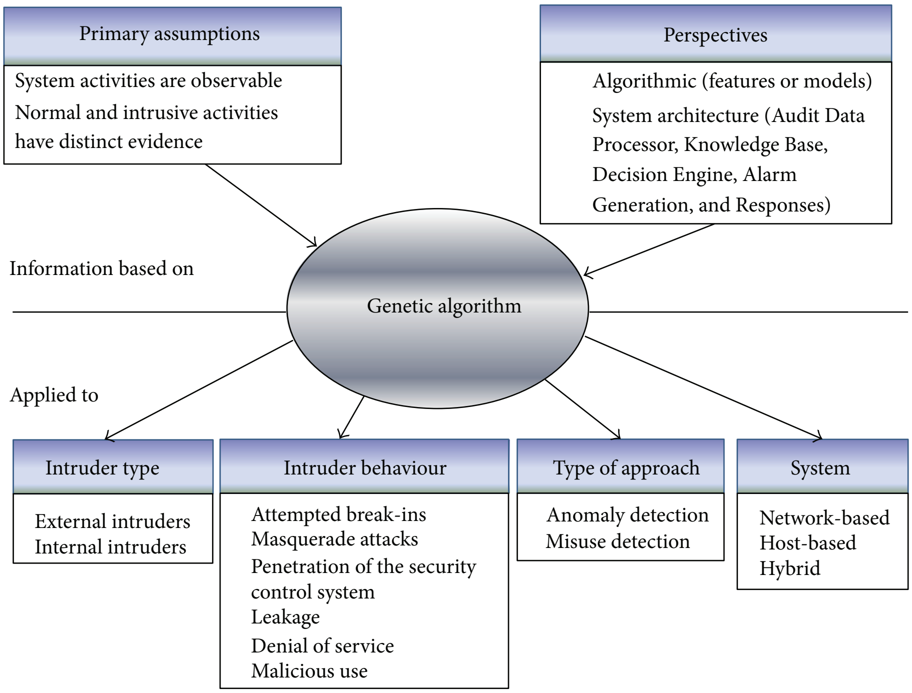


Figure 4: Intrusion detection systems and element types where genetic algorithms can be applied.

the perceived threat. A local monitoring node is introduced that acts like a proxy agent for the sink and is capable of monitoring neighbors. A Genetic algorithm based network IDS (GA-NIDS) is present in [17]. The proposed system considers many parameters such as protocol type, network services, and status of the connection to generate rules. The detection mechanism is trained on specific dataset, so that it can accurately identify and classify security attacks. In this mechanism, six rules are designed to detect six different types of denial of service (DoS) and probing attacks. The authors claim that the detection rate of DoS attacks is 100%.

Many IDSs are rule based where new and innovative attacks are not detected. An anomaly based IDS using concept of genetic algorithm is discussed in [18]. This framework uses set of classification rules which are derived from network audit data. It uses fitness function to monitor quality and stability of each rule.

Another evolutionary approach to intrusion detection is proposed in [19]. The proposed system is connected with a firewall, and it starts by capturing firewall entries. The data is then forwarded to genetic algorithm based system. The output of genetic algorithm is connected to an IDS. This mechanism is capable of detecting malicious connections and the detection rate is above 95%.

Table 3 presents comparison of few genetic algorithm based IDSs.

In our opinion, the detection mechanisms present in [17, 19] are not suitable for WSNs, as they demand more resources. The GA-NIDS considers many parameters such as protocol type, network services, and connection status for detection of an anomaly. Furthermore, different rules are designed for it which may consume more memory and energy due to computations involved in it.

Table 3: Comparison of genetic algorithm based IDSs.

IDS	Accuracy	Lightweight	Attack types
[16]	N/A	Yes	N/A
[17]	100%	No	DoS
[18]	N/A	Yes	Network layer
[19]	Above 95%	No	Malicious connections

The mechanism present in [19] takes input entries from firewall and forwards output entries to another IDS. Sensor nodes have limited processing capabilities, so they cannot operate with such complex mechanism for longer time.

When the genetic algorithm is applied to an IDS, several issues must be taken into account. The first one is the type of intrusion detection system purpose, and the second one is the element where it will be applied. Taking into account the classification provided in [20], we can draw Figure 4.

A general flow chart of a genetic algorithm for intrusion detection system will start using a randomly selected population of chromosomes. These chromosomes are encoded, on one hand, as observable system activities and evidence of normal and intrusive activities and, on the other hand, as the perspectives selected by the security administrator (Audit Data Processor, Knowledge Base, Decision Engine, Alarm Generation, and Responses). Figure 5 shows the general flow chart of a genetic algorithm for intrusion detection system.

5. Conclusion

Confidentiality, integrity, and availability of any network are of high importance. Network security is gaining importance

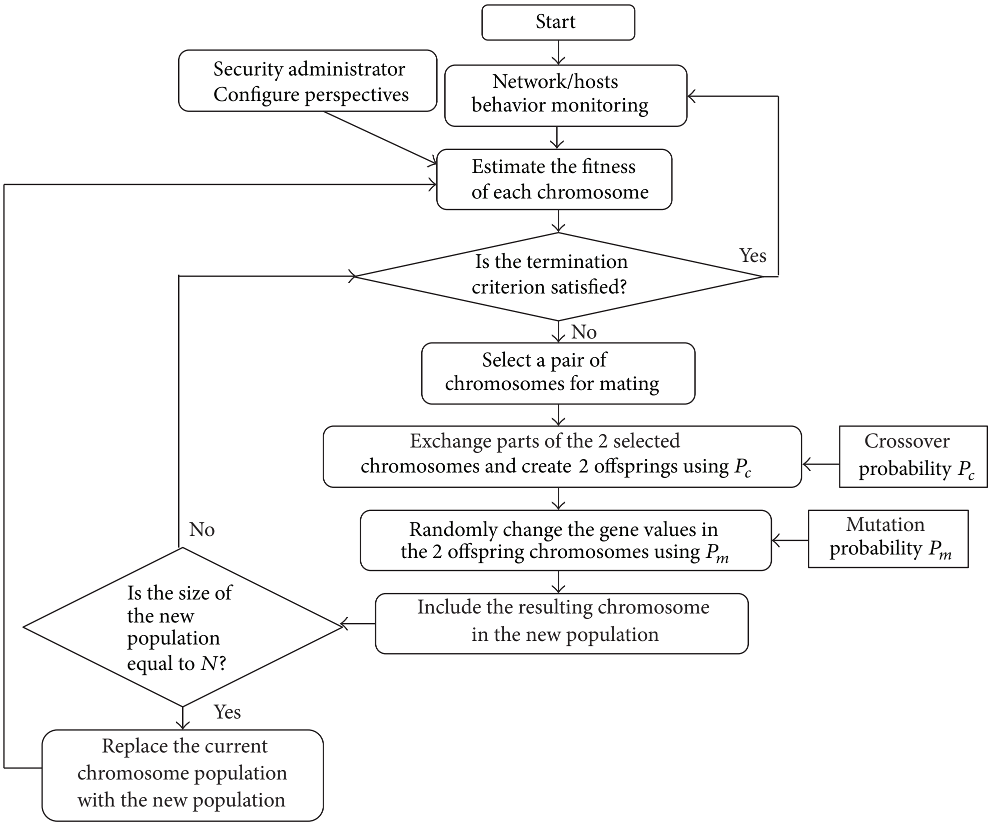

Start
Network/hosts behavior monitoring
Security administrator Configure perspectives
Estimate the fitness of each chromosome
Is the termination criterion satisfied?
Yes
No
Select a pair of chromosomes for mating
Exchange parts of the 2 selected chromosomes and create 2 offsprings using P_c
Crossover probability P_c
Randomly change the gene values in the 2 offspring chromosomes using P_m
Mutation probability P_m
Include the resulting chromosome in the new population
Is the size of the new population equal to N?
No
Yes
Replace the current chromosome population with the new population

Figure 5: Flow chart of a genetic algorithm for intrusion detection systems.

as the attackers introduce new threats and security vulnerabilities to compromise the normal operations of network [21–23]. WSNs are more vulnerable to different security attacks as they are deployed mostly in unattended environments. IDSs are capable of detecting intrusions and informing the professionals well in time. There are many methodologies and techniques which are used to design IDSs.

Research community is exploring biological concepts to design different mechansisms to solve computing related problems. Similarly, many biological concepts such as artificial immune system, artificial neural network, and genetic algorithm are used in IDSs.

This paper provides brief discription of bioinspired IDSs and their suitability for WSNs.

Acknowledgment

The authors extend their appreciation to the Distinguished Scientist Fellowship Program (DSFP) at King Saud University for funding this research.

References

[1] D. Bri, M. Garcia, J. Lloret, and P. Dini, "Real deployments of wireless sensor networks," in *Proceedings of the 3rd International Conference on Sensor Technologies and Applications (SENSORCOMM '09)*, pp. 415–423, Athens, Greece, June 2009.

[2] A. Radhika, D. Kavitha, and D. Haritha, "Mobile agent based routing in MANETS—attacks & defences," *Network Protocols and Algorithms*, vol. 3, no. 4, pp. 108–121, 2011.

[3] K. Sahadevaiah and P. V. G. D. Prasad Reddy, "Impact of security attacks on a new security protocol for mobile ad hoc networks," *Network Protocols and Algorithms*, vol. 3, no. 4, pp. 122–140, 2011.

[4] N. Alrajeh, S. Khan, and B. Shams, "Intrusion detection systems in wireless sensor networks: a review," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 167575, 7 pages, 2013.

[5] M. S. Sisodia and V. Raghuwanshi, "Anomaly base network intrusion detection by using random decision tree and random projection a fast network intrusion detection technique," *Network Protocols and Algorithms*, vol. 3, no. 4, pp. 93–107, 2011.

[6] N. A. Alrajeh, S. Khan, J. Lloret, and J. Loo, "Artificial neural network based detection of energy exhaustion attacks in wireless sensor networks capable of energy harvesting," *Ad Hoc & Sensor Wireless Networks*, vol. 2013, pp. 1–25, 2013.

[7] T. H. Lim, *Detecting anomalies in wireless sensor networks [Qualifying dissertation]*, Department of Computer Science, University of York, 2010.

[8] J. Kim, P. Bentley, C. Wallenta, M. Ahmed, and S. Hailes, "Danger is ubiquitous: detecting malicious activities in sensor networks using the dendritic cell algorithm," in *Artificial Immune Systems*, vol. 4163 of *Lecture Notes in Computer Science*, pp. 390–403, Springer, Berlin, Germany, 2006.

[9] C. Wallenta, J. Kim, P. J. Bentley, and S. Hailes, "Detecting interest cache poisoning in sensor networks using an artificial

immune algorithm," *Applied Intelligence*, vol. 32, no. 1, pp. 1–26, 2010.

[10] M. R. Ahmadi, "An intrusion prediction technique based on co-evolutionary im-mune system for network security (CoCo-IDP)," *International Journal of Network Security*, vol. 9, no. 3, pp. 290–300, 2009.

[11] M. Drozda, S. Schaust, and H. Szczerbicka, "AIS for misbehavior detection in wireless sensor networks: performance and design principles," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 3719–3726, September 2007.

[12] N. Alrajeh, S. Khan, J. Lloret, and J. Loo, "Artificial neural network based detection of energy exhaustion attacks in wireless sensor networks capable of energy harvesting," *Journal of Ad Hoc & Sensor Wireless Networks*, vol. 2013, pp. 1–25, 2013.

[13] Y. Y. Li and L. E. Parker, "Intruder detection using a wireless sensor network with an intelligent mobile robot response," in *Proceedings of the IEEE Southeast Conference*, pp. 37–42, April 2008.

[14] P. Mukherjee and S. Sen, "Using learned data patterns to detect malicious nodes in sensor networks," in *Distributed Computing and Networking*, vol. 4904 of *Lecture Notes in Computer Science*, pp. 339–344, Springer, Berlin, Germany, 2008.

[15] B. Abdullah, I. Abd-alghafar, G. I. Salama, and A. Abd-alhafez, "Performance evaluation of a genetic algorithm based approach to network intrusion detection system," in *Proceedings of the International Conference on Aerospace Sciences and Aviation Technology*, Military Technical College, Cairo, Egypt, 2009.

[16] R. Khanna, H. Liu, and H.-H. Chen, "Reduced complexity intrusion detection in sensor networks using genetic algorithm," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–5, June 2009.

[17] A. Goyal and C. Kumar, "GA-NIDS: a genetic algorithm based network intrusion detection system," 2008, http://www.cs.northwestern.edu/~ago210/ganids/GANIDS.pdf.

[18] A. A. Ojugo, A. O. Eboka, O. E. Okonta, R. E. Yoro, and F. O. Aghware, "Genetic algorithm rule-based intrusion detection system (GAIDS)," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 8, pp. 1182–1194, 2012.

[19] B. S. Dhak and S. Lade, "An evolutionary approach to intrusion detection system using genetic algorithm," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 2, pp. 632–637, 2012.

[20] T. S. Sobh, "Wired and wireless intrusion detection system: classifications, good characteristics and state-of-the-art," *Computer Standards and Interfaces*, vol. 28, no. 6, pp. 670–694, 2006.

[21] A. Triviño-Cabrera and S. Cañadas-Hurtado, "Survey on opportunistic routing in multihop wireless networks," *International Journal of Communication Networks and Information Security*, vol. 3, no. 2, pp. 170–177, 2011.

[22] D. P. Franco, F. D. Barboza, and N. M. Cardoso, "A secure method for authenticity verification of handwritten signatures through digital image processing and artificial neural networks," *International Journal of Communication Networks and Information Security*, vol. 5, no. 2, pp. 120–126, 2013.

[23] N. Jeyanthi, N. Ch. S. Iyengar, P. Kumar, and A. Kannammal, "An enhanced entropy approach to detect and prevent DDoS in cloud environment," *International Journal of Communication Networks and Information Security*, vol. 5, no. 2, pp. 110–129, 2013.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 760834, 13 pages
http://dx.doi.org/10.1155/2013/760834

Research Article

On Eavesdropping Attacks in Wireless Sensor Networks with Directional Antennas

Hong-Ning Dai,[1] Qiu Wang,[1] Dong Li,[1] and Raymond Chi-Wing Wong[2]

[1] *Faculty of Information Technology, Macau University of Science and Technology, Avenida Wai Long, Taipa, Macau*
[2] *Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong*

Correspondence should be addressed to Hong-Ning Dai; hndai@ieee.org

Received 2 March 2013; Revised 29 June 2013; Accepted 29 June 2013

Academic Editor: S. Khan

The eavesdropping attack is a serious security threat to a wireless sensor network (WSN) since the eavesdropping attack is a prerequisite for other attacks. Conventional WSNs consist of wireless nodes equipped with omnidirectional antennas, which broadcast radio signals in all directions and are consequently prone to the eavesdropping attacks. Different from omnidirectional antennas, directional antennas radiate radio signals on desired directions and potentially reduce the possibility of the eavesdropping attacks. In this paper, we propose a model to analyze the eavesdropping probability in both single-hop WSNs and multihop WSNs with omnidirectional antennas and directional antennas. We verify the correctness of our analytical model by conducting extensive simulations. We have found that using directional antennas in either single-hop WSNs or multihop WSNs can significantly reduce the eavesdropping probability. The reason of the improved security of WSNs with directional antennas lies in (i) *the smaller exposure region* of a directional antenna and (ii) *the fewer hops* to route a packet due to the longer transmission range of a directional antenna. Our results have also shown that the security improvement factor heavily depends on the node density, the antenna beamwidth, and the signal path loss factor.

1. Introduction

Recently, wireless sensor networks (WSNs) have received enormous interests from both industry and academia [1]. WSNs have been used in environmental monitoring, health care, surveillance security, farming, and so forth. Many studies assume that sensor nodes are deployed in random from an airplane. Those scatted sensor nodes self-organize to form an ad hoc network, in which data packets are transmitted through multihops from the source node to the destination node.

In WSNs, any wireless node residing in the *transmission* range of the transmitter can potentially decode the signal when both the transmitter and the receiver are unaware of the *reconnaissance* [2]. The reconnaissance, also named the *eavesdropping* activity, has attracted considerable attentions recently since many adversary attacks often follow the eavesdropping activity, for example, hear-and-fire attacks [3]. Specifically, there are two types of eavesdropping attacks in WSNs [4]: (i) *Passive Eavesdropping*, in which the *malicious* nodes detect the information by listening to the message transmission in the broadcasting wireless medium; (ii) *Active Eavesdropping*, where the malicious nodes actively grab the information via sending queries to transmitters by disguising themselves as friendly nodes. The study on the passive eavesdropping attacks is often more important than that on the active eavesdropping attacks since the malicious nodes must have the knowledge of the friendly nodes via conducting passive eavesdropping activities before they can actively attack the friendly nodes. Thus, we only consider the passive eavesdropping attacks in this paper.

Conventional WSNs typically consist of nodes equipped with omnidirectional antennas which broadcast radio signals uniformly in all directions. Only a portion of these signals can reach the destinations and most of them are lost. This property of radiating signals omnidirectionally inevitably leads to *high interference* and a *short transmission range*. Both these two factors severely limit the network performance of WSNs equipped with omnidirectional antennas. We call such networks as wireless omnidirectional sensor networks (*WONs*).

Compared with omnidirectional antennas, directional antennas can concentrate most of radio signals on desired directions. In other undesired directions, there are no radio signals or the weakened signals. Therefore, using directional antennas in WSNs can potentially reduce the interference [5]. Besides, the transmission range can be significantly extended compared with omnidirectional antennas. We call such networks as directional-antennas wireless sensor networks (*DAWNs*).

1.1. Contributions. In this paper, we only concentrate on the *passive eavesdropping attack* in *WONs* and *DAWNs*. We have found that using directional antennas in WSNs can significantly improve the network security in terms of reducing the *eavesdropping probability* in both single-hop networks and multihop networks. Our contributions are summarized as follows.

The first contribution of this paper is to formally establish the eavesdropping model in WSNs with consideration of omnidirectional antennas and directional antennas. In particular, we propose *the exposure region* to determine whether an adversary node can eavesdrop the transmission or not. We also define the eavesdropping condition, the single-hop eavesdropping probability, and the multihop eavesdropping probability.

Secondly, we analyze the eavesdropping attacks in both single-hop networks and multihop networks with both omnidirectional antennas and directional antennas. We have found that a *DAWN* has a much lower eavesdropping probability than a *WON* in either single-hop networks or multihop networks. The security improvement of directional antennas owes to the *smaller exposure region* and the *fewer hops to route a packet*.

Last, we conduct extensive simulations to verify the correctness of our analytical models. We show that the simulation results exactly agree with our analytical model in both single-hop *WONs* and single-hop *DAWNs*. We also show that both the simulation results and the analytical results in multihop networks keep the same trend in the eavesdropping probability although there exist quite small gaps between them. We have found the reasons behind this effect and pointed out the future direction.

The remainder of the paper is organized as follows. Section 2 presents the models and the definitions. In Section 3, we analyze the eavesdropping attack in single-hop networks. Section 4 presents the analytical results of the eavesdropping attach in multihop networks. We then discuss the simulation results in Section 5. Section 6 reviews the related work. Finally, we conclude the paper in Section 7.

2. Models and Notations

We adopt the notations shown in (Notation) throughout the paper. Sections 2.1 and 2.2 present the directional antenna model and the transmission model, respectively. In Section 2.3, we propose an eavesdropping model to analyze the eavesdropping attacks.

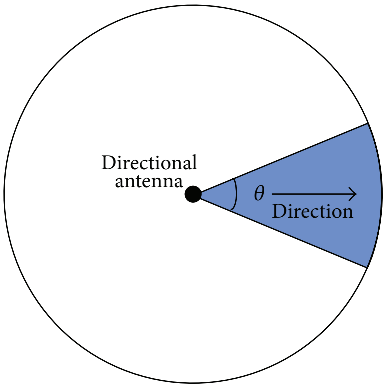


FIGURE 1: The antenna model.

2.1. Antenna Model. In this paper, we consider a directional antenna model that was used in previous studies [6–11]. This model can simplify our analysis. The reasons why we choose the model are summarized as follows. Firstly, even in a more realistic model, the sidelobes and backlobes are so small that they can be ignored. For example, in a more realistic model (the cone-sphere model) [12] in which the sidelobes and backlobes are counted, the gain of the main lobe is more than 100 times of the gain of the sidelobes. Secondly, smart antennas often have null capability that can almost eliminate the sidelobes and backlobes.

We assume that a directional antenna gain G_d is within a specific angle θ, where θ is the beamwidth of the antenna, as shown in Figure 1. The gain outside the beamwidth is assumed to be zero. More specifically, we have

$$G_d = \begin{cases} \dfrac{2\pi}{\theta} & \text{within } \theta \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The antenna gain of an omnidirectional antenna can be regarded as a special case in our model when the beamwidth $\theta = 2\pi$. Then, we have $G_o = 1$.

Note that a directional antenna generally has a beamwidth $\theta < \pi$. Therefore, we have $G_d > G_o$. Moreover, the narrower the beamwidth of a directional antenna is, the higher antenna gain it has.

2.2. Channel Model. We next describe the channel model. We denote the transmission power of node i by P_i and represent the channel gain from node i to node j as γ_{ij}. We also assume that all nodes use the same transmission power P_t, that is, $P_i = P_t$. Thus, the received power at node j is $P_t \cdot \gamma_{ij}$. The signal-to-interference-plus-noise ratio at node j denoted by SINR is defined to be

$$\frac{P_t \cdot \gamma_{ij}}{\eta + \sum_{k \neq i} P_k \cdot \gamma_{kj}}. \tag{2}$$

The transmission from node i can be successfully received by node j if and only if

$$\text{SINR} \geq \beta, \tag{3}$$

where β is the minimum signal to interference and noise ratio and η is the environmental noise power level, which is assumed to be the same for all nodes.

Note that there are only one transmitter and one receiver in a single-hop network and all other nodes are *adversary* nodes, which are passive eavesdroppers and will not transmit actively. In a multihop network (see Section 4), there are N *good* nodes and M passive *adversary* nodes, which will not transmit actively. Besides, we also assume that only one of all the N good nodes can transmit at a time. Based on these assumptions on a single-hop network and a multihop network, we can concentrate our analysis on the eavesdropping attacks and ignore the interference from other nodes. Therefore, we have $\sum_{k \neq i} P_k \cdot \gamma_{kj} = 0$. We then have

$$\text{SINR} = \frac{P_t \cdot \gamma_{ij}}{\eta} \geq \beta. \tag{4}$$

In this paper, we consider the large-scale path loss in the channel model [13]. Thus, the channel gain is given by

$$\gamma_{ij} = C \cdot G_t \cdot G_r \cdot \frac{1}{d_{ij}^{\alpha}}, \tag{5}$$

where d_{ij} denotes the distance between node i and node j, $C = (\lambda/4\pi)^2$ (λ is the wavelength of the signal), G_t and G_r are the antenna gains for the transmitter and the receiver, respectively, and α is the path loss factor ($2 \leq \alpha \leq 4$) [13].

Note that the channel model also holds for both the normal transmission and the eavesdropping attack. When the channel model is used for the normal transmission, G_t and G_r in (5) are the antenna gain of a transmitter and the antenna gain of a receiver, respectively, where a transmitter and a receiver are also denoted as *Good* nodes. When the channel model is used for the eavesdropping attack, G_t and G_r in (5) are the antenna gain of a transmitter and the antenna gain of an adversary node, respectively, where an adversary node is also denoted as a *malicious* or *bad* node.

2.3. Eavesdropping Model. We next formally propose our eavesdropping model. First, we map *WONs* and *DAWNs* to the following two cases, which are summarized in Table 1.

Table 1: Directional case and omnidirectional case.

	Omnidirectional (I)	Directional (II)
Transmitter	Omni	Directional
Receiver	Omni	Omni
Adversary	Omni	Omni

In Case I (Omnidirectional), the transmitters, the receivers, and the adversary nodes use omnidirectional antennas (Omni for short). The Omnidirectional case corresponds to a *WON*. In Case II (Directional), the transmitters use directional antennas while the receivers and adversary nodes use omnidirectional antennas. The Directional case corresponds to a *DAWN*. Note that the adversary nodes are equipped with omnidirectional antennas in both Cases I and II. This is because, due to the directionality of a directional antenna, the adversary nodes have no knowledge about the position of the transmitter. They have to listen *omnidirectionally* in order to eavesdrop the messages. Similarly, we also consider that the receivers use omnidirectional antennas so that they can find the transmitters easily.

Second, we analyze the channel model for the eavesdropping attacks. If an adversary node can correctly decode the information from the transmitter, the SINR at the adversary node must satisfy the condition given in Inequality (4). After combining Inequality (4) and (5), we have

$$d_{ij} \leq \left(\frac{C \cdot P_t \cdot G_t \cdot G_r}{\beta \cdot \eta}\right)^{1/\alpha}. \tag{6}$$

We denote the right-hand side (RHS) of (6) as

$$R_{\max} = \left(\frac{C \cdot P_t \cdot G_t \cdot G_r}{\beta \cdot \eta}\right)^{1/\alpha} \tag{7}$$

which is the maximum radius within which an adversary node can correctly eavesdrop the information from the transmitter.

We then define the *exposure* region as follows.

Definition 1 (exposure region). The exposure region of a transmitter is an area that any adversary nodes within this area can potentially eavesdrop the transmission from the transmitter.

It is obvious that the area of the exposure region is determined by the geometric shape of the exposure region and the maximum radius $R_{\max}$, which is defined in (7). Note that in Case I (Omnidirectional), a transmitter has a *circular* exposure region with radius $R_{\max} = R_o$, which can be calculated by

$$R_o = \left(\frac{C \cdot P_t \cdot G_o^2}{\beta \cdot \eta}\right)^{1/\alpha}, \tag{8}$$

where G_o^2 means that both the transmitter and the adversary node are using omnidirectional antennas.

In Case II (Directional), a transmitter has an exposure region of a *sector* with angle θ and radius $R_{\max} = R_d$, which is given by

$$R_d = \left(\frac{C \cdot P_t \cdot G_d \cdot G_o}{\beta \cdot \eta}\right)^{1/\alpha}, \tag{9}$$

where G_d means that the transmitter is using a directional antenna and G_o means that the adversary node is using omnidirectional antenna.

Note that both the receiver and the adversary nodes have the same minimum SINR (i.e., β) to decode the information to the transmitter. Figures 2 and 3 illustrate the exposure

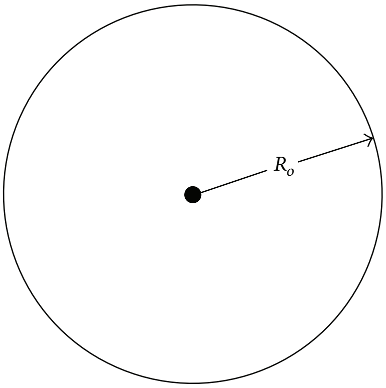


Figure 2: The exposure region of an omnidirectional antenna.

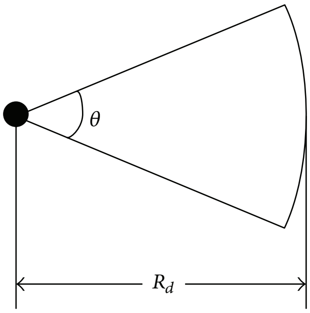


Figure 3: The exposure region of a directional antenna.

regions of a directional antenna and an omnidirectional antenna, respectively.

We then define the *eavesdropping condition* as follows.

Definition 2 (eavesdropping condition). An adversary node can successfully eavesdrop the information from the transmitter *if and only if the adversary node falls into the exposure region of the transmitter*.

To evaluate the seriousness of eavesdropping attacks, we define the *eavesdropping* probability $p(\mathrm{e})$ of a WSN in single-hop networks and multihop networks, respectively.

Definition 3 (single-hop eavesdropping probability). The eavesdropping probability $p(\mathrm{e})$ equals the probability that at least one adversary node falls into the exposure region of the transmitter.

Definition 4 (multihop eavesdropping probability). The eavesdropping probability of a multihop transmission is the probability that at least one-hop transmission is eavesdropped.

It is obvious that $0 \leq p(\mathrm{e}) \leq 1$. In order to compare the eavesdropping attacks at two different types of WSNs, namely, WSN_A and WSN_B, we define the security improvement factor as follows.

Definition 5 (security improvement factor). The security improvement factor $I_{A/B}$ of WSN_A over WSN_B is the ratio of the eavesdropping probability of WSN_A to the eavesdropping probability of WSN_B, that is, $I_{A/B} = p_A(\mathrm{e})/p_B(\mathrm{e})$, where $p_A(\mathrm{e})$ denotes the eavesdropping probability of WSN_A and $p_B(\mathrm{e})$ denotes the eavesdropping probability of WSN_B.

When $I_{A/B} > 1$, it implies that WSN_B has the lower eavesdropping probability than WSN_A, that is, WSN_B is more secure than WSN_A. Similarly, $I_{A/B} < 1$ means that WSN_A is more secure than WSN_B. When $I_{A/B} = 1$, WSN_A has the eavesdropping security equal to that of WSN_B. Note that the definition of Security Improvement Factor applies for both single-hop networks and multihop networks.

Finally, we describe the node distribution. In this paper, we consider that both the adversary nodes and the good nodes are distributed in a two-dimensional plane. We use a Poisson point process to model the distribution of the nodes [14]. In particular, the probability $p(i)$ of finding i nodes in an area of S is given by

$$p(i) = f(i, S) = \frac{(\rho S)^i}{i!} e^{-\rho S}, \tag{10}$$

where $f(i, S)$ is the probability mass function and ρ is the node density.

3. Security Analysis on Single-Hop Networks

In this section, we analyze the eavesdropping probabilities of Omnidirectional case and Directional case in single-hop networks, in which all packets are transmitted through only one hop.

3.1. Eavesdropping Probability of WSNs. Generally, the eavesdropping probability of a single-hop WSN can be obtained by the following lemma.

Lemma 6. *The eavesdropping probability can be calculated by*

$$p(\mathrm{e}) = 1 - e^{-\rho \cdot S}. \tag{11}$$

Proof. From the definition of the eavesdropping probability, that is, Definition 3, we have

$$\begin{aligned} p(\mathrm{e}) &= p(i \geq 1) \\ &= 1 - p(i < 1) = 1 - p(i = 0), \end{aligned} \tag{12}$$

where i denotes the number of adversary nodes falling into the exposure region of a transmitter. □

Since the distribution of adversary nodes follows Poisson point process as defined in (10), we then have

$$p(\mathrm{e}) = 1 - e^{-\rho \cdot S}. \tag{13}$$

As shown in Lemma 6, to determine the eavesdropping probability, we need to calculate the area of the exposure region S first. We then calculate the exposure region S under the Omnidirectional case (*WONs*) and the Directional case (*DAWNs*), respectively.

In Omnidirectional case, the transmitter, the receiver, and the adversary nodes are equipped with omnidirectional antennas. The exposure region of an omnidirectional antenna is a circle with radius R_o, as shown in Figure 2. The area of the exposure region can be calculated by

$$S_o = \pi \cdot R_o^2, \tag{14}$$

where R_o can be obtained by (8).

In Directional case, the transmitter is equipped with a directional antenna while eavesdroppers are equipped with omnidirectional antennas. The exposure region of a directional antenna is a sector with radius R_d and angle θ, as shown in Figure 3. The area of the exposure region can be calculated by

$$S_d = \frac{\theta}{2\pi} \cdot \pi \cdot R_d^2, \tag{15}$$

where R_d can be obtained by (9).

After replacing S in (11) by S_o (defined in (14)) and S_d (defined in (15)), respectively, we obtain the eavesdropping probability of the Omnidirectional case and the eavesdropping probability of the Directional case, which are given as the following lemma.

Lemma 7. *The eavesdropping probability of the Omnidirectional case, denoted by P_o, can be calculated by*

$$p_o(\mathrm{e}) = 1 - e^{-\rho \cdot S_o}. \tag{16}$$

The eavesdropping probability of the Directional case, denoted by P_d, can be calculated by

$$p_d(\mathrm{e}) = 1 - e^{-\rho \cdot S_d}. \tag{17}$$

3.2. Comparison between WONs and DAWNs under Single-Hop Networks. To simplify the analysis, we define the *reference node density N_o*, which is the average number of nodes within an exposure region of the Omnidirectional case, as follows:

$$N_o = \rho \cdot S_o = \rho \cdot \pi \cdot R_o^2. \tag{18}$$

We then have the following theorem to compare a *WON* with a *DAWN* in terms of the security improvement factor.

Theorem 8. *The security improvement factor of a WON over a DAWN is equal to*

$$I_{o/d} = \frac{1 - e^{-N_o}}{1 - e^{-N_o \cdot (\theta/2\pi)^{1-(2/\alpha)}}}. \tag{19}$$

Proof. From the definition of security improvement factor, that is, Definition 5, we have

$$I_{o/d} = \frac{p_o(\mathrm{e})}{p_d(\mathrm{e})}, \tag{20}$$

where $p_o(\mathrm{e})$ and $p_d(\mathrm{e})$ denote the eavesdropping probabilities of a *WON* and a *DAWN*, respectively.

Replacing $p_o(\mathrm{e})$ and $p_d(\mathrm{e})$ in (20) by RHS of (16) and RHS of (17), respectively, we have

$$\begin{aligned} I_{o/d} &= \frac{p_o(\mathrm{e})}{p_d(\mathrm{e})} \\ &= \frac{1 - e^{-\rho \cdot \pi \cdot R_o^2}}{1 - e^{-\rho \cdot \pi \cdot R_d^2 \cdot (\theta/2\pi)}} \\ &= \frac{1 - e^{-N_o}}{1 - e^{-N_o \cdot (\theta/2\pi)^{1-(2/\alpha)}}}. \end{aligned} \tag{21}$$

□

We then analyze the security improvement factor of a *WON* over a *DAWN*. In particular, we have the following.

Corollary 9. *DAWNs have the eavesdropping probability no higher than that of WONs under the same network settings. More precisely, one has the following.*

(i) *When the pass loss factor $\alpha = 2$, $I_{o/d} = 1$, no matter what θ and N_o are, it implies that a DAWN has the same eavesdropping probability as a WON.*

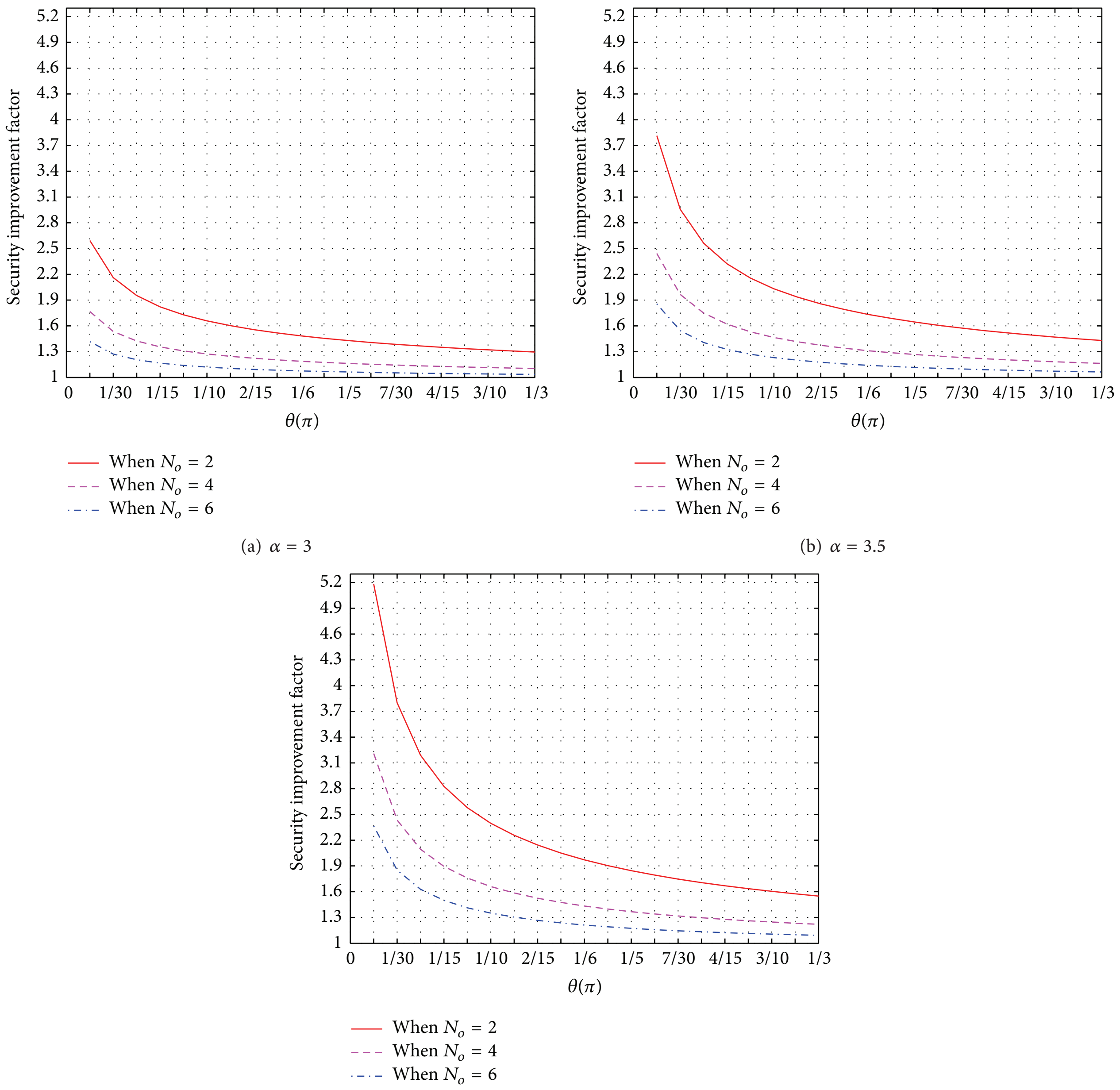
(ii) *When the pass loss factor $\alpha > 2$, $I_{o/d} > 1$, it implies that a DAWN has higher eavesdropping probability than a WON.*

We then calculate the security improvement factor $I_{o/d}$ with varied beamwidth θ, node density N_o, and path loss factor α. Figure 4 illustrates the security improvement factor of single-hop transmissions.

It is shown in Figures 4(a), 4(b), and 4(c) that the security improvement factor $I_{o/d}$ is always greater than 1 when $\alpha > 2$. In other words, using directional antennas in such environments can always reduce the eavesdropping probability and improve the network security. Figure 4 also shows that the security improvement factor $I_{o/d}$ increases when the path loss factor α increases. Therefore, using directional antennas in higher path-loss environments may improve the network security further.

Besides, it is also shown in Figure 4 that with the increased node density N_o, the security improvement factor $I_{o/d}$ decreases. This is because more adversary nodes fall into the exposure region with the increased node density. Therefore, from the adversary point of view, the best way to improve the success rate of eavesdropping attacks is to distribute more adversary nodes in the whole network.

Moreover, Figure 4 also shows that the security improvement factor $I_{o/d}$ increases with the decreased antenna beamwidth θ. In other words, the narrower antenna beamwidth leads to the higher security improvement. For example, in Figure 4(c), $I_{o/d} = 2.83$ when $\alpha = 4$, $N_o = 2$, and $\theta = \pi/15$ (i.e., 12°). Thus, in a network distributed with dense adversary nodes, we should use narrow-beam antennas to avoid eavesdropping.



(a) $\alpha = 3$

(b) $\alpha = 3.5$

(c) $\alpha = 4$

Figure 4: Security improvement factor of single-hop transmissions.

4. Security Analysis on Multihop Networks

In this section, we extend our analysis from single-hop networks to multihop networks. We first derive the eavesdropping probability of multihop transmissions. Then, we analyze the security improvement factor of *WONs* over *DAWNs* under multihop networks.

4.1. Routing Path in Multihop Networks. To analyze the eavesdropping probability of multihop transmissions, we construct a simple routing scheme that chooses a route with the shortest distance to forward data packets. We first introduce the Source-Destination (S-D) Line model [3, 15].

In the S-D Line model, we divide the unit-area plane into a lot of equal-sized square cells as shown in Figure 5. Each of them has an identical area of $a(n)$. The size of the cell, $a(n)$ should be greater enough to ensure that there is at least one node in each cell. It is the necessary condition to ensure that the network is connected.

In this S-D Line model, we directly draw a line to connect a source node S and its destination node D. Then, node S will send data packets to its destination D by multihop forwarding those packets along the cells lying on its S-D line. Figure 5 shows an example of S-D lines, where the green line indicates a S-D Line. In the case of *WONs*, the packets are forwarded along the *adjacent* cells lying on the S-D line. For example,

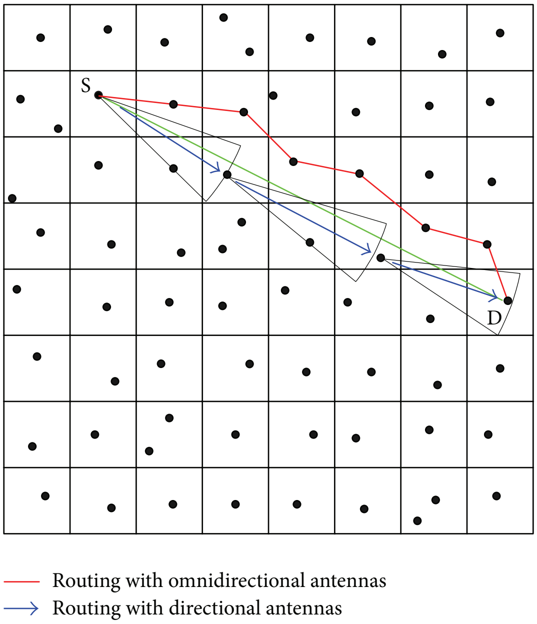

S
D
Routing with omnidirectional antennas
Routing with directional antennas

Figure 5: The S-D Line model.

the red line as shown in Figure 5 denotes the routing path from S to D in a *WON*. However, the cells lying on the S-D line in a *DAWN* are not necessary to be adjacent since a directional antenna has a longer transmission range than an omnidirectional antenna. As shown in Figure 5, only 3 hops are needed from S to D, compared with the omnidirectional antenna case, which requires 7 hops from S to D. Therefore, using directional antennas can potentially reduce the number of hops.

We then calculate the number of hops required to route a packet from S to D. Since calculating the exact number of hops is difficult, we are only concerned about the number of hops H from S to D. In the S-D line model, the number of hops depends on both the length of the S-D line l and the transmission range of each hop, which is R_o in a *WON* and is R_d in a *DAWN*.

We next calculate the number of hops H_o of a *WON*, which is bounded by

$$H_o = \left[\frac{l}{R_o}\right], \tag{22}$$

where $[\cdot]$ denotes the near integer function and R_o is the maximum transmission range of an omnidirectional antenna, which can be calculated from (8).

Similarly, the number of hops H_d of a *DAWN* is bounded by

$$H_d = \left[\frac{l}{R_d}\right], \tag{23}$$

where R_d is the maximum transmission range of a directional antenna, which can be calculated from (9).

4.2. Eavesdropping Probability of Multihop Networks. In general, the eavesdropping probability of a multihop WSN (either *WON* or *DAWN*) can be obtained by the following lemma.

Lemma 10. *The eavesdropping probability of multihop networks can be calculated by*

$$p_m(\mathrm{e}) = 1 - (1 - p(\mathrm{e}))^H, \tag{24}$$

where H is the number of hops and $p(\mathrm{e})$ is the eavesdropping probability of a single-hop transmission, which can be calculated by Lemma 6.

Proof. From the definition of the eavesdropping probability of multihop networks, that is, Definition 4, we have

$$p_m(\mathrm{e}) = 1 - (1 - p(\mathrm{e}))^H. \tag{25}$$

Note that we assume that each hop has the same eavesdropping probability. □

For a *WON*, since $p(\mathrm{e}) = p_o(\mathrm{e})$, which can be calculated from (16) and the number of hops $H = H_o$, which can be calculated from (22). Therefore, we have the eavesdropping probability of a *WON* denoted by p_{mo}

$$p_{mo}(\mathrm{e}) = 1 - (1 - p_o(\mathrm{e}))^{H_o}. \tag{26}$$

For a *DAWN*, since $p(\mathrm{e}) = p_d(\mathrm{e})$, which can be calculated from (17) and the number of hops $H = H_d$, which can be calculated from (23), we have the eavesdropping probability of a *DAWN* denoted by p_{md}

$$p_{md}(\mathrm{e}) = 1 - (1 - p_d(\mathrm{e}))^{H_d}. \tag{27}$$

From (24), we have found that with the increased number of hops H, the multihop eavesdropping probability also significantly increases. Since directional antennas have a longer transmission range than omnidirectional antennas, that is, $R_d > R_o$, the number of required hops for directional antennas is smaller than that for omnidirectional antennas. Therefore, using directional antennas in multihop networks can potentially reduce the multihop eavesdropping probability, which will be verified in Section 5.

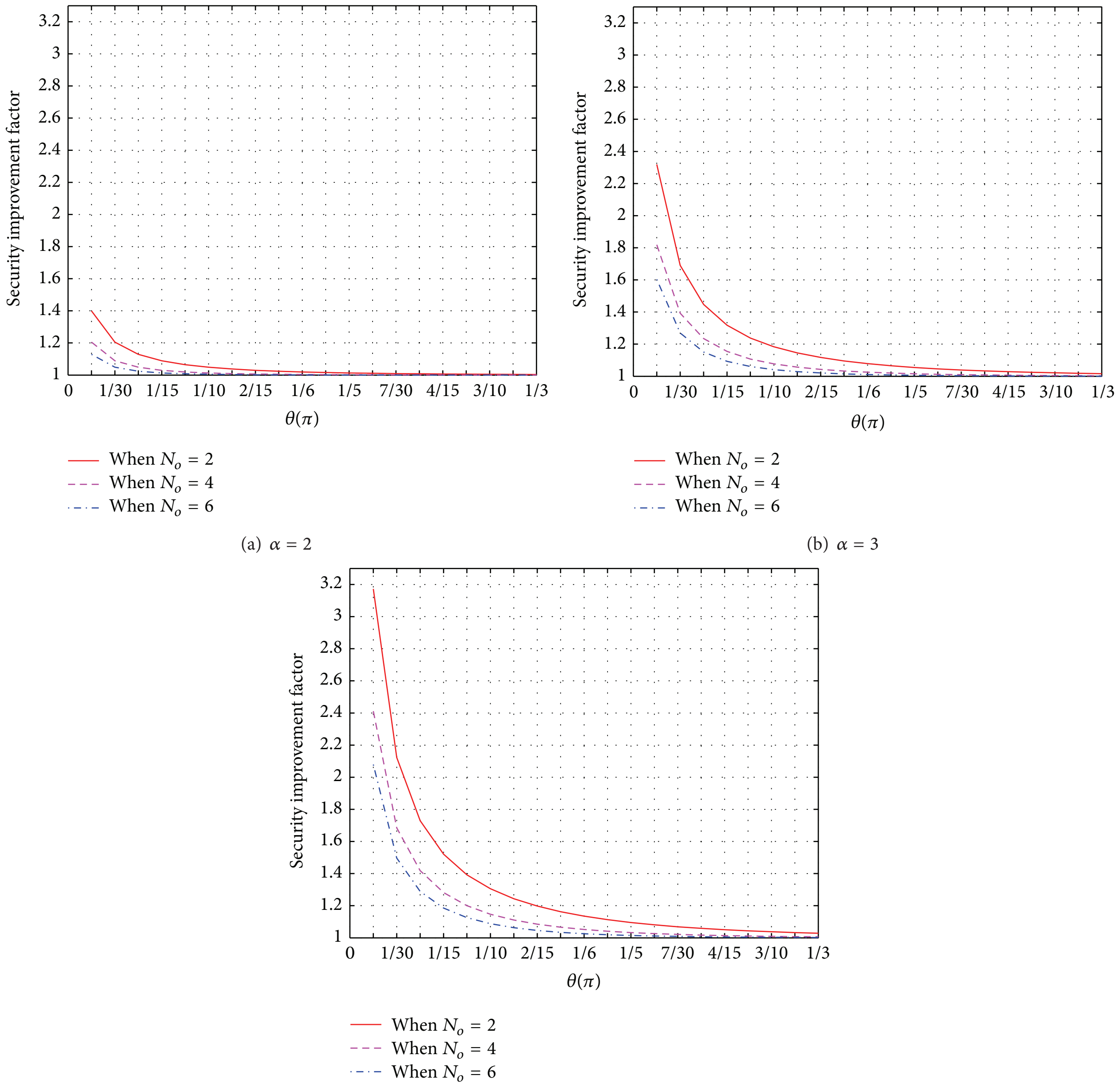
4.3. Comparison between WONs and DAWNs under Multihop Networks. We then have the following theorem to compare a *WON* with a *DAWN* in terms of the security improvement factor under multihop networks.

Theorem 11. *The security improvement factor of a WON over a DAWN under multihop networks is equal to*

$$I_{o/d}^m = \frac{1 - (1 - p_o(\mathrm{e}))^{H_o}}{1 - (1 - p_d(\mathrm{e}))^{H_d}}. \tag{28}$$

Proof. From the definition of security improvement factor, that is, Definition 5, we have

$$I_{o/d}^m = \frac{p_{mo}(\mathrm{e})}{p_{md}(\mathrm{e})}. \tag{29}$$



(a) $\alpha = 2$

(b) $\alpha = 3$

(c) $\alpha = 4$

Figure 6: The security improvement factor of multihop transmissions.

After replacing p_{mo} and p_{md} of (28) by RHS of (26) and RHS of (27), we have

$$I_{o/d}^{m} = \frac{1 - \left(1 - p_o\left(\mathrm{e}\right)\right)^{H_o}}{1 - \left(1 - p_d\left(\mathrm{e}\right)\right)^{H_d}}. \quad (30)$$

□

Note that when $H_o = 1$ and $H_d = 1$, $I_{o/d}^{m} = p_o(\mathrm{e})/p_d(\mathrm{e})$, which is equal to $I_{o/d}$, given by (19). Thus, the security improvement factor $I_{o/d}$ under the single-hop networks can be regarded as a special case of the multihop networks.

We then calculate $I_{o/d}^{m}$ with varied beamwidth θ, node density N_o, and path loss factor α. Figure 6 illustrates security improvement factor $I_{o/d}^{m}$. It is shown in Figure 6 that the security improvement factor $I_{o/d}^{m}$ decreases when the node density N_o increases (e.g., N_o increases from 2 to 6). This is because, when the node density N_o increases, more adversary nodes fall into the exposure region, which results in higher eavesdropping probability. Similar to single-hop networks, the security improvement factor of multihop transmissions $I_{o/d}^{m}$ also significantly increases with the decreased antenna beamwidth θ. In other words, the narrower antenna beamwidth θ is, the higher security improvement $I_{o/d}^{m}$ is.

Two factors contribute to the increment of the security improvement of multihop transmissions: (i) the smaller exposure region with narrower beamwidth θ leads to the less eavesdropping probability of single-hop transmissions; (ii) the narrower beamwidth θ is, the higher antenna gain is (refer to (1)), which also results in the longer transmission range R_d and consequently leads to the less eavesdropping probability of multihop transmissions $p_{md}(\mathrm{e})$.

Moreover, it is also shown in Figure 6 that there is a higher multihop security improvement factor $I^{m}_{o/d}$ in a higher path-loss environment (e.g., $\alpha = 4$) than that in a lower path-loss environment (e.g., $\alpha = 3$). Using directional antennas in such environments may bring more benefits.

In addition, different from the single-hop networks, the security improvement $I^{m}_{o/d}$ under the multihop networks is always greater than 1 even when the path loss factor $\alpha = 2$ (note that $I_{o/d} = 1$ when $\alpha = 2$ under the single-hop networks as shown in Corollary 9). This mainly owes to the reduced number of hops by using directional antennas. Our analytical results imply that multihop networks with directional antennas are generally more secure than multihop networks with omnidirectional antennas.

5. Empirical Results

In this section, we conduct extensive simulations to evaluate the correctness and the accuracy of our proposed models in single-hop networks (Section 5.1) and multihop networks (Section 5.2). In the simulations, both adversary nodes and good nodes are randomly distributed on a plane of area $l \times l\,\mathrm{m}^2$. To eliminate the border effects, we use the sub-area approach [16]. Specifically, we only compute the eavesdropping probability of the nodes within an inner square of area $l' \times l'\,\mathrm{m}^2$, where l' is sufficiently smaller than l. Besides, the simulation results are calculated by averaging over 10000 random topologies in single-hop networks and obtained by averaging over 1000 random topologies in multihop networks.

5.1. Single-Hop Networks. Figure 7 shows the eavesdropping probability versus node density when $\alpha = 3$ (Figure 7(a)) and $\alpha = 4$ (Figure 7(b)). Note that the analytical results are represented in curves (ana) and the simulation results are indicated by markers (sim) in Figure 7. Besides, each simulation result is calculated by averaging over 10000 random topologies.

As shown in Figures 7(a) and 7(b), the simulation results almost exactly match the analytical results in all cases. Besides, it is shown in Figures 7(a) and 7(b) that the eavesdropping probability increases when N_o increases. This is because, when N_o increases, the more adversary nodes fall into the exposure regions, leading to the higher eavesdropping probability (the networks become the less secure).

Moreover, we can see in Figure 7 that a *DAWN* always has the lower eavesdropping probability than a *WON* in all cases, which further confirms our earlier observations in Section 3. In other words, using directional antennas in wireless networks can potentially improve the security. In addition, when α is fixed, we can see that the eavesdropping probability decreases with the increased beamwidth θ. It implies that a narrower-beam antenna can potentially reduce the eavesdropping probability and further improve the security of transmissions.

Furthermore, we can also see that the eavesdropping probability of a *DAWN* significantly drops with the increased path loss factor α (e.g., α increases from 3 in Figure 7(a) to 4 in Figure 7(b)) while α just slightly affects the eavesdropping probability of a *WON*. It is shown in [13] that the path loss factor α is generally greater than 3 in urban outdoor environments. Therefore, using directional antennas in such environments may bring more benefits on reducing the eavesdropping probability than using omnidirectional antennas.

5.2. Multihop Networks. Tables 2 and 3 show the eavesdropping probabilities under multihop networks when $\alpha = 3$ and $\alpha = 4$, respectively. Note that we choose lower density N_o (i.e., ranging from 0.05 to 0.5) in both Tables 2 and 3. The main reason lies in the higher eavesdropping probability under multihop networks than that under single-hop networks (it is obvious that the whole transmission is eavesdropped once one-hop transmission is eavesdropped). Besides, we choose $l = 1200\,\mathrm{m}$ in Table 2 and $l = 400\,\mathrm{m}$ in Table 3. This is because we have to limit the number of hops when $\alpha = 4$ in order to avoid that the eavesdropping probability reaches one too fast. More specifically, when $\alpha = 4$, both R_d and R_o drop significantly, resulting in the increased number of hops (recall that H_d and H_o depend on R_d and R_o) and consequently leading to the severely increased eavesdropping probability. Furthermore, to obtain the proper routing path, which approximates that derived under the S-D Line model (Section 4), we first choose a source-destination node pair, which spans nearly the dimension of the network (i.e., $\approx l$). Then we obtain the shortest routing path based on the Dijkstra's algorithm [17]. Note that each of our simulation results is calculated by averaging over 1000 random topologies. (Note that it is extremely time consuming to obtain the multihop result for each random topology.)

Both Tables 2 and 3 show that a *WON* has much higher eavesdropping probability than a *DAWN* when $\alpha = 3$ and $\alpha = 4$. Besides, the eavesdropping probability of a *WON* goes more quickly to reach one than that of a *DAWN*. The reason behind this phenomenon lies the higher single-hop eavesdropping probability of a *WON* than that of a *DAWN*.

As shown in Tables 2 and 3, the eavesdropping probability $p_{mo}(\mathrm{e})$ of a *WON* and the eavesdropping probability $p_{md}(\mathrm{e})$ of a *DAWN* increase significantly with the increased node density. This agrees with our observation that the higher N_o is, the more adversary nodes fall into the exposure regions, which results in the higher single-hop eavesdropping probability, either $p_o(\mathrm{e})$ or $p_d(\mathrm{e})$. As a result, the multihop eavesdropping probabilities $p_{mo}(\mathrm{e})$ and $p_{md}(\mathrm{e})$ increase.

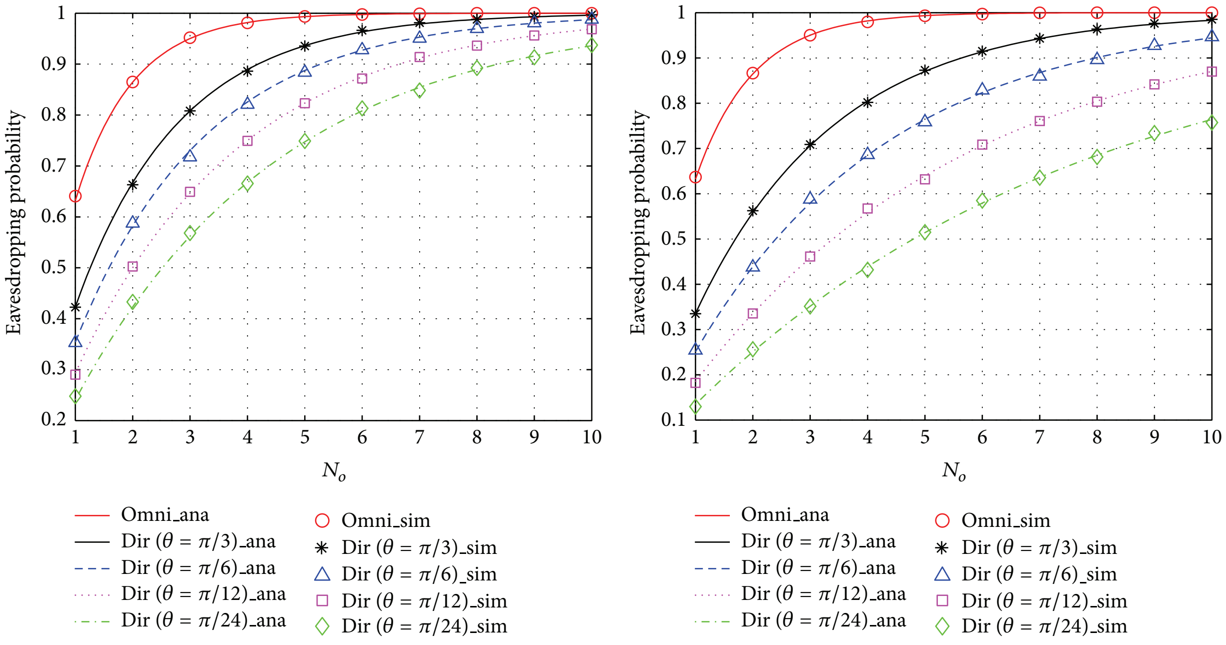
Besides, Tables 2 and 3 also show that the narrower the beamwidth θ is, the smaller the eavesdropping probability $p_{md}(\mathrm{e})$ is. For example, $p_{md}(\mathrm{e}) = 0.544$ (simulation) with beamwidth $\theta = \pi/12$ is much smaller than $p_{md}(\mathrm{e}) = 0.714$ (simulation) with beamwidth $\theta = \pi/6$ with the node density $N_o = 0.20$ and the path loss factor $\alpha = 3$. This mainly owes to the two factors described in Section 4. Moreover, it is also shown in Tables 2 and 3 that $p_{md}(\mathrm{e})$ decreases when the path loss factor α increases. Therefore, using directional antennas in such high path loss environments may gain more security improvement.

Table 2: Eavesdropping probability versus node density N_o when $\alpha = 3$.

N_o	$p_{mo}(\mathrm{e})$		$p_{md}(\mathrm{e})$ $(\theta = \pi/3)$		$p_{md}(\mathrm{e})$ $(\theta = \pi/6)$		$p_{md}(\mathrm{e})$ $(\theta = \pi/12)$	
	Analytical	Simulation	Analytical	Simulation	Analytical	Simulation	Analytical	Simulation
0.05	0.727	0.624	0.338	0.360	0.231	0.255	0.144	0.187
0.10	0.925	0.851	0.562	0.605	0.408	0.444	0.268	0.346
0.15	0.979	0.944	0.710	0.751	0.544	0.589	0.374	0.439
0.20	0.994	0.988	0.808	0.866	0.649	0.714	0.464	0.544
0.25	0.998	0.989	0.873	0.915	0.730	0.790	0.542	0.607
0.30	0.999	0.996	0.916	0.943	0.792	0.835	0.608	0.685
0.35	0.999	0.996	0.944	0.953	0.840	0.890	0.664	0.746
0.40	0.999	0.999	0.963	0.973	0.877	0.884	0.713	0.818
0.45	0.999	0.998	0.976	0.990	0.905	0.928	0.754	0.829
0.50	0.999	0.999	0.984	0.990	0.927	0.941	0.790	0.863

Table 3: Eavesdropping probability versus node density N_o when $\alpha = 4$.

N_o	$p_{mo}(\mathrm{e})$		$p_{md}(\mathrm{e})$ $(\theta = \pi/3)$		$p_{md}(\mathrm{e})$ $(\theta = \pi/6)$		$p_{md}(\mathrm{e})$ $(\theta = \pi/12)$	
	Analytical	Simulation	Analytical	Simulation	Analytical	Simulation	Analytical	Simulation
0.05	0.683	0.570	0.264	0.300	0.171	0.176	0.106	0.120
0.10	0.899	0.801	0.458	0.488	0.313	0.374	0.201	0.236
0.15	0.968	0.903	0.601	0.663	0.431	0.482	0.286	0.305
0.20	0.989	0.963	0.706	0.756	0.528	0.554	0.362	0.399
0.25	0.996	0.980	0.784	0.818	0.609	0.655	0.430	0.483
0.30	0.998	0.993	0.841	0.878	0.676	0.712	0.490	0.554
0.35	0.999	0.999	0.883	0.909	0.731	0.766	0.544	0.604
0.40	0.999	0.999	0.913	0.933	0.777	0.820	0.593	0.651
0.45	0.999	0.999	0.936	0.959	0.815	0.828	0.636	0.671
0.50	0.999	0.999	0.953	0.961	0.847	0.873	0.674	0.752



(a) $\alpha = 3$

(b) $\alpha = 4$

Figure 7: Eavesdropping probability versus node density N_o, averaged over 10000 random-generated topologies with $l = 1200$ m (curves = analytical results and markers = simulation results), where omni and dir represent *DAWNs* and *WONs*, respectively.

There exist quite small gaps between the analytical values and the simulation results as shown in Tables 2 and 3 though both of them have the same trend. More precisely, as shown in Tables 2 and 3 that the analytical values are always greater than the simulation results in a *WON* while the analytical values are always smaller than the simulation results in a *DAWN*. This effect may owe to two factors: (1) the number of hops in simulations is often greater than that in analysis; (2) there are some overlapping regions counted in each hop when we conduct simulations. It is obvious that Factor (1) will lead to the higher simulation results than the analytical values while Factor (2) will lead to the lower simulation results than the analytical values. In a *WON*, Factor (2) dominates Factor (1) since the overlapping ratio of the circular region is very high. As a result, the analytical value is always slightly greater than the simulation result. On the contrary, in a *DAWN*, Factor (1) dominates Factor (2), consequently leading to the higher simulation result than the analytical value. A more precise analytical model for multihop networks is expected to be proposed in the future while it is beyond the scope of this paper.

6. Related Work

Wireless Sensor Networks (WSNs) are prone to the malicious attacks due to the shared wireless medium, the multihop transmissions, and the decentralized control scheme [2, 18–20]. In a WSN, any wireless node residing in the *transmission* range of the transmitter can potentially decode the signal when both the transmitter and the receiver are unaware of the *reconnaissance*. Besides, it is also difficult to implement the centralized control mechanisms in WSNs. Furthermore, multihop communications are also suggested in WSNs to reduce the interference and to improve the network capacity [21]. However, the multihop communication is vulnerable to the malicious attacks.

One of the malicious attacks, namely, the *eavesdropping* attack, has attracted considerable attentions recently since many other malicious attacks often follow the eavesdropping activity. As summarized in [4, 18], there are two types of eavesdropping attacks in WSNs: passive eavesdropping and active eavesdropping. The study on the passive eavesdropping attacks is often more important than that on the active eavesdropping attacks since it is a prerequisite that the malicious nodes have the knowledge of the good nodes via conducting passive eavesdropping activities.

There are a number of studies on investigating the passive eavesdropping attack [2–4, 18]. But, most of them considered a *WON*, in which each node is equipped with an *omnidirectional* antenna, which radiates the radio signals in all directions and consequently is more prone to the eavesdropping attacks. Compared with omnidirectional antennas, directional antennas can concentrate radio signals on desired directions. In other undesired directions, there are no radio signals or weakened signals. Thus, using directional antennas in WSNs can potentially reduce the interference and consequently improve the network performance.

There are a number of studies on using directional antennas in wireless ad hoc networks. The first category of them mainly focuses on the theoretical analysis on the network performance, for example, the network capacity and the transmission delay. In particular, studies [6, 22, 23] derived the approximated network capacity of wireless networks with directional antennas, in which each node is equipped with only one directional antenna and only one channel is used. More specifically, Yi et al. [6] show that using directional antenna in arbitrary networks achieves a capacity gain of $2\pi/\sqrt{\alpha\beta}$ when both the transmitter and the receiver are equipped with directional antennas, where α and β are transmitter and receiver antenna beamwidth, respectively. Under random networks (in which n nodes are randomly placed, directional antennas of each node are randomly adjusted, and the destination of a flow is also randomly chosen), the throughput improvement factor is $4\pi^2/\alpha\beta$ when both the transmitter and the receiver are equipped with directional antennas. Other studies [10, 11] are focused on multi-channel and multi-interface networks with directional antennas, which are proved to have a higher network capacity than that single-channel networks with directional antennas. Besides, the study [24] shows that the transmission delay in wireless networks with directional antennas due to multihop transmissions can be significantly reduced due to the longer transmission range of directional antennas. The second category of studies focus on improving the network performance in Medium Access Control (MAC) layer [8, 12, 25–37]. In particular, using directional antennas in wireless networks often results in the new hidden terminal problem and the deafness problem, which were first addressed in [28]. Both the new hidden terminal problem and the deafness problem severely degrade the network performance. Therefore, a number of studies were proposed to address them [8, 30–32, 34, 35]. However, many of these solutions only solve either the new hidden terminal problem or the deafness problem but not both. Besides, many of them often have additional overheads due to sending additional control packets. For example, circular DMAC [8] needs transmitting multiple RTS frames for each data packet.

Most of the above studies focus on improving the network performance by using directional antennas. However, there is little work on the security issue by using directional antennas. The study [38] is one of the earliest studies on exploring using directional antennas in wireless networks to improve the network security. It is shown in [38] that using directional antennas can significantly reduce the average detection probability compared with using omnidirectional antennas. However, their studies only analyze single-hop multihop transmission and do not consider other benefits of directional antennas, such as the longer transmission range, which may reduce the number of hops and consequently improve the security further.

7. Conclusion

In this paper, we have explored using directional antennas in wireless sensor networks to improve the network security in terms of reducing the eavesdropping probability. In particular, we analyzed the eavesdropping probability of single-hop networks and that of multihop networks. We have found that using directional antennas in either a single-hop network or a multihop network can significantly reduce the eavesdropping probability. The security improvements of using directional antennas owe to the *smaller exposure region* and the *fewer hops* due to the longer transmission range. Besides, we also derived the security improvement factors of single-hop transmissions and multihop transmissions. We have found that both the single-hop security improvement factor and the multihop security improvement factor heavily depend on the antenna beamwidth, the node density, and the path loss factor. It is shown that using a narrow beam antenna can significantly improve the network security by reducing the eavesdropping probability.

There are some interesting topics in the eavesdropping activities of *DAWNs*. For example, most of current studies always assume that the adversary nodes are uniformly and randomly distributed in the networks. What about the eavesdropping probability if the distribution of the adversary nodes is non-uniform and deliberate? Besides, does the power control schemes will affect the eavesdropping probability of a *DAWN*?

Notations

G_t:	Antenna gain of transmitters
G_r:	Antenna gain of receivers
G_d:	Directional antenna gain
G_o:	Omnidirectional antenna gain
θ:	Antenna beamwidth, that is, the angle between the half-power points of the main lobe
γ_{ij}:	The channel gain from node i to node j
P_t:	Fixed transmission power of all nodes
SINR:	Signal-to-Interference-Plus-Noise Ratio
α:	Signal path loss factor
β:	Minimum signal to interference and noise ratio
η:	Fixed environmental noise power level
R_d:	Maximum radius of the exposure region of directional antennas
R_o:	Maximum radius of the exposure region of omnidirectional antennas
ρ:	Node density
N_o:	The average number of nodes in a circle with radius R_o (the reference node density)
$p(\mathrm{e})$:	Single-hop eavesdropping probability
$p_d(\mathrm{e})$:	Single-hop eavesdropping probability of a directional antenna
$p_o(\mathrm{e})$:	Single-hop eavesdropping probability of an omnidirectional antenna
$p_m(\mathrm{e})$:	Multihop eavesdropping probability
$p_{md}(\mathrm{e})$:	Multihop eavesdropping probability of a directional antenna
$p_{mo}(\mathrm{e})$:	Multihop eavesdropping probability of an omnidirectional antenna
$I_{A/B}$:	The security improvement factor of WSN_A over WSN_B.

Acknowledgments

The work described in this paper was supported by Macao Science and Technology Development Fund under Grant no. 036/2011/A and Grant no. 081/2012/A3. The authors would like to thank Guodong Han and Qinglin Zhao for their valuable comments.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] F. Anjum and P. Mouchtaris, *Security for Wireless Ad Hoc Networks*, Wiley-Interscience, New York, NY, USA, 1st edition, 2007.

[3] J. C. Kao and R. Marculescu, “Eavesdropping minimization via transmission power control in Ad-Hoc wireless networks,” in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON ’06)*, pp. 707–714, Reston, Va, USA, September 2006.

[4] M. Anand, Z. G. Ivesy, and I. Leez, “Quantifying eavesdropping vulnerability in sensor networks,” in *Proceedings of the 2nd International Workshop on Data Management for Sensor Networks (DMSN ’05)*, pp. 3–9, 2005.

[5] X. Huang, J. Wang, and Y. Fang, “Achieving maximum flow in interference-aware wireless sensor networks with smart antennas,” *Ad Hoc Networks*, vol. 5, no. 6, pp. 885–896, 2007.

[6] S. Yi, Y. Pei, and S. Kalyanaraman, “On the capacity improvement of Ad Hoc wireless networks using directional antennas,” in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ’03)*, pp. 108–116, 2003.

[7] L. Bao and J. J. Garcia-Luna-Aceves, “Transmission scheduling in Ad Hoc networks with directional antennas,” in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom ’02)*, pp. 48–58, September 2002.

[8] T. Korakis, G. Jakllari, and L. Tassiulas, “A MAC protocol for full exploitation of directional antennas in ad-hoc wireless networks,” in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ’03)*, pp. 98–107, June 2003.

[9] Z. Zhang, “Pure directional transmission and reception algorithms in wireless Ad Hoc networks with directional antennas,” in *Proceedings of the IEEE International Conference on Communications (ICC ’05)*, vol. 5, pp. 3386–3390, Seoul, Republic of Korea, May 2005.

[10] H. N. Dai, K. W. Ng, R. C. W. Wong, and M. Y. Wu, “On the capacity of multi-channel wireless networks using directional antennas,” in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM ’08)*, pp. 1301–1309, Phoenix, Ariz, USA, April 2008.

[11] J. Zhang and X. Jia, “Capacity analysis of wireless mesh networks with omni or directional antennas,” in *Proceedings of the IEEE INFOCOM*, pp. 2881–2885, Rio de Janeiro, Brazil, April 2009.

[12] R. Ramanathan, “On the performance of Ad Hoc networks with beamforming antennas,” in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc ’01)*, pp. 95–105, Long Beach, Calif, USA, 2001.

[13] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2002.

[14] Y. Wang and J. J. Garcia-Luna-Aceves, "Directional collision avoidance in Ad Hoc networks," *Performance Evaluation*, vol. 58, no. 2-3, pp. 215–241, 2004.

[15] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-delay trade-off in energy constrained wireless networks," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, March 2004.

[16] C. Bettstetter, "On the connectivity of Ad Hoc networks," *Computer Journal*, vol. 47, no. 4, pp. 432–447, 2004.

[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Boston, Mass, USA, 3rd edition, 2009.

[18] N. A. Alrajeh, S. Khan, and B. Shams, "Intrusion detection systems in wireless sensor networks: a review," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 167575, 7 pages, 2013.

[19] N. A. Alrajeh, S. Khan, J. Lloret, and J. Loo, "Secure routing protocol using cross-layer design and energy harvesting in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 374796, 11 pages, 2013.

[20] N. Meghanathan, "A survey on the communication protocols and security in cognitive radio networks," *International Journal of Communication Networks and Information Security*, vol. 5, pp. 19–38, 2013.

[21] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[22] A. Spyropoulos and C. S. Raghavendra, "Capacity bounds for Ad-Hoc networks using directional antennas," in *Proceedings of the IEEE International Conference on Communications (ICC '03)*, pp. 348–352, May 2003.

[23] J. Zhang and S. C. Liew, "Capacity improvement of wireless Ad Hoc networks with directional antennae," *Mobile Computing and Communications Review*, vol. 10, no. 4, pp. 17–19, 2006.

[24] H. N. Dai, "Throughput and delay in wireless sensor networks using directional antennas," in *Proceedings of the 5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP '09)*, pp. 421–426, Melbourne, Australia, December 2009.

[25] Y. B. Ko, V. Shankarkumar, and N. H. Vaidya, "Medium access control protocols using directional antennas in Ad Hoc networks," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 1, pp. 13–21, Tel Aviv, Israel, March 2000.

[26] A. Nasipuri, S. Ye, J. You, and R. E. Hiromoto, "A MAC protocol for mobile Ad Hoc networks using directional antennas," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '2000)*, vol. 3, pp. 1214–1219, Chicago, Ill, USA, September 2000.

[27] M. Takai, J. Martin, R. Bagrodia, and A. Ren, "Directional virtual carrier sensing for directional antennas in mobile Ad Hoc networks," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '02)*, pp. 183–193, Lausanne, Switzerland, June 2002.

[28] R. R. Choudhury, X. Yang, N. H. Vaidya, and R. Ramanathan, "Using directional antennas for medium access control in Ad Hoc networks," in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom '02)*, pp. 59–70, Atlanta, Ga, USA, September 2002.

[29] Z. Huang, C. C. Shen, C. Srisathapornphat, and C. Jaikaeo, "A busy-tone based directional MAC protocol for Ad Hoc networks," in *Proceedings of the MILCOM*, vol. 2, pp. 1233–1238, October 2002.

[30] R. R. Choudhury and N. H. Vaidya, "Deafness: a MAC problem in Ad Hoc networks when using directional antennas," in *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP '04)*, pp. 283–292, October 2004.

[31] H. Gossain, C. Cordeiro, and D. P. Agrawal, "MDA: an efficient directional MAC scheme for wireless Ad Hoc networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '05)*, pp. 3633–3637, St. Louis, Mo, USA, December 2005.

[32] H. Gossain, C. Cordeiro, and D. P. Agrawal, "Minimizing the effect of deafness and hidden terminal problem in wireless Ad Hoc networks using directional antennas," *Wireless Communications and Mobile Computing*, vol. 6, no. 7, pp. 917–931, 2006.

[33] M. Takata, M. Bandai, and T. Watanabe, "A receiver-initiated directional MAC protocol for handling deafness in Ad Hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, pp. 4089–4095, Istanbul, Turkey, July 2006.

[34] J. Wang, H. Zhai, P. Li, Y. Fang, and D. Wu, "Directional medium access control for Ad Hoc networks," *Wireless Networks*, vol. 15, no. 8, pp. 1059–1073, 2009.

[35] H. N. Dai, K. W. Ng, and M. Y. Wu, "A busy-tone based MAC scheme for wireless Ad Hoc networks using directional antennas," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 4969–4973, Washington, DC, USA, November 2007.

[36] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, and P. Steenkiste, "DIRC: increasing indoor wireless capacity using directional antennas," in *Proceedings of the ACM SIGCOMM Conference on Data Communication (SIGCOMM '09)*, pp. 171–182, August 2009.

[37] H. N. Dai, K. W. Ng, and M. Y. Wu, "On busy-tone based MAC protocol for wireless networks with directional antennas," *Wireless Personal Communications*, 2013.

[38] X. Lu, F. Wicker, P. Lio, and D. Towsley, "Security estimation model with directional antennas," in *Proceedings of the IEEE Military Communications Conference (MILCOM '08)*, pp. 1–6, San Diego, Calif, USA, November 2008.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 205920, 16 pages
http://dx.doi.org/10.1155/2013/205920

Research Article

Detection and Prevention of Selective Forwarding-Based Denial-of-Service Attacks in WSNs

Youngho Cho and Gang Qu

Department of Electrical and Computer Engineering and Institute for Systems Research, University of Maryland, College Park, 20742, USA

Correspondence should be addressed to Gang Qu; gangqu@umd.edu

Received 2 May 2013; Accepted 18 June 2013

Academic Editor: S. Khan

Designing wireless sensor networks (WSNs) that can work reliably in the presence of inside packet drop attackers is very challenging. Current trust mechanisms and avoidance approaches are promising but have their limitations. Avoidance approaches transmit multiple copies of the packets to avoid attackers and cause high overhead. In trust mechanisms, each sensor monitors it neighbors, evaluates their trustworthiness, classifies them as either trustworthy or untrustworthy, and then discards untrustworthy sensors from the network. However, malicious insiders, which are legitimate members of the network and know exactly what their monitoring nodes know, can launch attacks carefully to avoid being detected and discarded from the network. In this paper, we first show that this is possible by introducing a selective forwarding-based denial-of-service (DoS) attack. We then propose an enhanced trust mechanism to detect such attackers and identify their victims. Furthermore, we design two attacker-aware protocols to reroute victim nodes' packets by avoiding the attackers. We conduct extensive OPNET simulations to validate our claims and demonstrate the advantages of our proposed approaches. Finally, as a complementary defensive method to our detection and avoidance approaches, we introduce a prevention routing algorithm that proactively prevents the attack and provide our preliminary results to evaluate its performance.

1. Introduction

In wireless sensor networks (WSNs), sensor nodes will generate data packets and send them to the base station (BS) in a multihop collaborative fashion due to their limited energy and transmission range. While being routed to the BS, data packets may be lost from collision, congestion, noise, or other network problems. The so-called *insider packet drop attacks* refer to a set of attacks where compromised nodes intentionally drop packets [1]. Such attackers disguise their malicious behavior behind the aforementioned natural packet loss phenomenon. This type of attack has become a serious security threat in WSNs [1–3]. A well-positioned malicious insider can be on the routing path of many sensor nodes and thus receive many data packets. It can simply drop them to cause damage to the network.

Selective forwarding attack, where the attacker drops only some packets and at some arbitrary time, is the most difficult insider packet drop attack to defend against [2]. Normally such an attacker seeks to achieve one of the following two goals. First, degrade the performance of the network in terms of packet loss rate. Second, prevent data collected by certain sensor nodes from reaching the BS. In the second case, the victim node will not be able to talk to the BS, and we name this attack *selective forwarding-based denial-of-service (DoS) attack*. Most reported studies on selective forwarding attacks focus on the detection of the attacker with the first goal [4–7]. As we will discuss later, these approaches are not effective against selective forwarding-based DoS attacks.

As a motivation for the importance of studying selective forwarding-based DoS attacks, we consider a WSN deployed in a territory for intruder detection. With the help of insiders that perform the selective forwarding-based DoS attack, an intruder will be able to enter the territory from the area monitored by victim nodes (to the selective forwarding-based DoS attacks) without being noticed by the BS. When the intruder can communicate with the inside attackers, they can launch the *synchronized insider-outsider colluding DoS attack* so the insider attackers can target different victims at different times, and the intruder can explore the territory covered by the victim nodes only.

Trust mechanism has been proven as a promising approach to identify inside packet drop attackers [3, 8–12]. In such approach, each node will monitor its neighbor's packet forwarding behavior and use this observation to measure the trustworthiness of its neighbors. Once a neighbor's trust value falls below a predetermined threshold, the monitoring node will consider this neighbor as an inside attacker and eliminate it from the routing table.

Another conceptually different approach to defend against insider packet drop attacks is avoidance [2, 13], where multiple copies of the packets are sent to the BS through multiple disjoint paths. As long as there is a path that does not contain any attackers, the packets will be delivered to the BS successfully. However, this approach has a very high cost in terms of network traffic, transmission energy, and so forth [2, 7].

In this paper, we study the selective forwarding-based DoS attacks and propose effective detection and avoidance mechanisms as well as a prevention routing algorithm to defend against such attacks. Specifically,

(i) we first describe a simple selective forwarding-based DoS attack and show that the popular trust-based approaches (such as beta [14] and entropy [10] trust mechanisms) for inside attacker detection fail to detect such attack. We also analyze the potential damage this attack can cause to the network,

(ii) we then propose a *source-level trust evaluation scheme* to enhance the beta and entropy trust mechanisms for effective detection of the selective forwarding-based DoS attackers. Once the attacker is identified, we propose two *avoidance strategies* to reroute the victim's packets so they can reach the BS,

(iii) we validate our claims and evaluate the performance of our detection and avoidance mechanisms with extensive OPNET simulations,

(iv) as a complementary defensive mechanism to our detection and avoidance methods, we also introduce a prevention routing algorithm to proactively prevent the selective forwarding-based DoS attacks and show our preliminary results to evaluate its performance.

For simplicity, during the discussion of the threats and detection of insider packet drop attacks, we do not consider natural packet drops caused by network problems. However, our simulation settings include lossy networks and the natural packet drops due to that network problems will be reported.

The rest of this paper is organized as follows. Section 2 covers related work on insider packet drop attacks and the current countermeasures. Then in Section 3, we describe a selective forwarding-based DoS attack that none of the current defending approaches can detect to motivate our work. We propose our detection and avoidance approaches in Section 4 and evaluate their performance in the packet routing domain in Section 5. In Section 6, as a complementary defensive mechanism to our detection and avoidance methods, we introduce a prevention routing algorithm where an attacker has to choose between "not attacking" and "attacking and being caught." We conclude this paper in Section 7.

2. Related Work

Attackers to a network can be insiders, outsiders, or both. WSNs deployed for security applications (such as monitoring in the battlefield) are normally equipped with cryptography-based authentication and authorization mechanisms to prevent outside attackers from launching eavesdropping or packet modification. Thus outsider attacks are limited to direct physical damage of sensors or jamming the communication channel [15]. However, inside attackers have many advantages [2, 12, 15]. First, they are legitimate members of the network and will not be caught by authentication or authorization. Second, inside attackers can disrupt network operations by modifying packet information or dropping critical packets. Finally, inside attackers can collude with outside attackers to cause more severe damage to the network as we have described in the introduction [16].

Inside attackers can launch various types of attacks actively (such as modification, packet drop, or misrouting) or passively (such as eavesdropping). Among these, packet drop attacks not only can cause significant network performance degradation, but also cannot be prevented by authentication and authorization [2]. Below are three representative types of insider packet drop attacks [2, 8, 10].

Blackhole Attacks. The blackhole attacker drops all received packets. It will cause the most serious damage to the network among all types of packet drop attacks during the same amount of time. However, it can be easily captured by the monitoring neighbors as it consistently drops all their packets.

On-Off Attack. When attack is on, the attacker drops all received packets, then forwards all received packets when attack is off, and repeats this drop-forward pattern periodically. This attacker can appear suspicious to its neighbor during its attack period when it acts like blackhole attacks and can also be detected easily when the attack on period is long or the on-off pattern is discovered.

Selective Forwarding Attacks. As we described in the introduction, such attackers can either drop packets randomly or selectively. It is much more challenging to defend these attacks than blackhole and on-off attacks.

Current defending approaches against selective forwarding attacks are either *detection approach* or *avoidance approach*. The detection approaches will fail to detect the attacker and victims in our proposed selective forwarding-based DoS attack. The avoidance approaches will solve the problem, but it is very expensive and may not suit for WSN applications where each sensor has limited resource.

Most of the reported efforts focused on random selective forwarding attacks [4–7]. For example, Hai and Huh [4] presented a neighbor-based monitoring and detection mechanism using two-hop neighbor knowledge where each exchanges its one-hop neighbors' packet forwarding behavior periodically. However, this approach introduces network overhead due to periodic information exchange between nodes and is vulnerable to false information provided by malicious neighbors. In the multihop acknowledgement scheme [7], each node in the forwarding path is responsible for detecting attackers. Specifically, some randomly chosen nodes (called ACK nodes) will report ACKs back to the source node (hop by hop) using the same but reversed routing path when they receive a packet. However, this approach has several problems. First, it is unclear how to locate the exact attacker. Second, their detection scheme depends on other nodes' observations, and thus their scheme is vulnerable to false accusation from malicious neighbors. The trust mechanisms with watchdog, as we have discussed earlier, solve these problems by monitoring whether the next node in the routing path forwards the packets or not [3, 9–11, 17]. Despite its many known limitations, the trust mechanism has been a promising solution to defend against insider packet drop attacks.

Instead of detecting the attackers, the avoidance approaches focus on how to deliver the packets successfully with the existence of the attackers. A popular way to achieve this is to use multipath routing paths [2, 13, 18, 19]. In [2], the authors pointed out that k disjoint multipath routing can completely defend against selective forwarding attacks with no more than $k-1$ compromised nodes. However, the multipath routing approach has a couple of drawbacks [7]. First, communication overhead significantly increases as the number of paths increases, and thus it may lead to increase collision and interference. As a result, the packet delivery performance of a routing can be dramatically degraded. Second, since this approach cannot catch and discard the attackers, this approach can be compromised if an adversary locates at least one attacker in each routing path. Similarly, a multiple data flow scheme using multiple disjoint topologies was introduced in [13]. In this scheme, a sending node sends its packets through one or more randomly chosen topologies among the preestablished multiple topologies to mitigate selective forwarding attacks.

3. A Selective Forwarding-Based DoS Attack

In this section, we first describe the current trust mechanisms and trust-based routing approaches to avoid inside attackers. Then we introduce a selective forwarding-based DoS attack and show that the current trust mechanisms fails to detect such attack.

3.1. Trust Mechanism.

3.1. Trust Mechanism. A trust mechanism defines a trust value (or trustworthiness) for each sensor node, and how each node measures the trustworthiness of its neighbors. It detects insider packet drop attacks in the following three stages.

Neighbor Behavior Monitoring. Each node monitors and records its neighbors' behavior such as packet forwarding. Watchdog [3] is a popular monitoring mechanism used in this stage. Each node M records all of its recently forwarded packets in a buffer. When M sends a packet to its neighbor node A, M monitors whether A forwards the packet toward the BS by overhearing A's packet transmission. Then, each overheard packet will be compared with the packet sent to A. When a match is found, M records that A has forwarded the packet and removes it from the buffer. If a packet remains in the buffer for a period longer than a predetermined time, the watchdog considers that A failed to forward the packet. In this paper, we use this watchdog mechanism in the OPNET network simulator [20].

Trust Measurement. Based on the data collected in the previous stage, a trust model will measure the trustworthiness of the node being monitored [8, 10, 14]. For example, when a node is observed to have forwarded the packet s times and dropped the packet f times, the beta trust model [14] will assign this node a trust value using the following formula:

$$T_{\text{Beta}} = \frac{s+1}{s+f+2}. \tag{1}$$

The entropy trust model [10] uses entropy function:

$$H(p) = -p\log_2 p - (1-p)\log_2(1-p), \tag{2}$$

where p is the trust value in beta trust model, and define the trust value by

$$T_{\text{Entr}} = \begin{cases} 1 - H(p), & \text{for } 0.5 \le p \le 1; \\ H(p) - 1, & \text{for } 0 \le p < 0.5. \end{cases} \tag{3}$$

Note that in (1), the trust value is between 0 and 1. But the trust value in (3) is between −1 and 1. To have a nonnegative trust value between 0 and 1, we define

$$T^*_{\text{Entr}} = \frac{1 + T_{\text{Entr}}}{2}. \tag{4}$$

Detection. By comparing the measured trust value with a predetermined threshold Θ_T, a node can decide whether its neighbor is trustworthy. If the neighbor's trust value is less than Θ_T, it will be considered as an inside attacker. Depending on the network's trust mechanism, the detection of inside packet drop attackers may or may not be broadcast to the rest of the nodes in the network. In this paper, we assume that the decision will not be broadcast for simplicity.

3.2. Trust-Based Routing. We use the popular greedy perimeter stateless routing (GPSR) [21] as an example to show how a trust mechanism can help to detect and avoid inside attackers.

Consider the WSN with 20 nodes shown in Figure 1. Node 3 relays the packets from nodes 4, 5, 9, 10, 15, and its own packets to node 2 which will then send them to the BS based on GPSR (depicted by solid lines with arrowhead). In a trust mechanism, node 3 will use its watchdog to monitor node 2.

When node 2 drops packets from node 3, node 3 will reevaluate the trust value of node 2. If the trust value falls below the threshold value, node 3 will treat node 2 as an

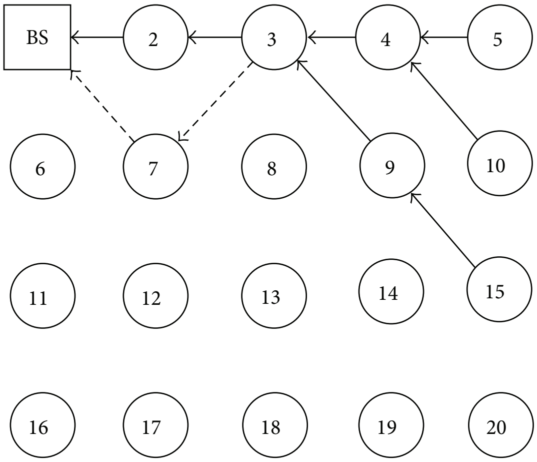


Figure 1: GPSR (solid lines) and trust-based GPSR (dotted lines).

inside attacker. A trust-based routing algorithm will then find a new routing path to avoid node 2. In this case, node 3 will forward the packets to node 7, hoping that node 7 will deliver the packets to the BS (the dotted lines with arrowheads in Figure 1).

Many researchers [9, 10, 17, 22–24] have shown that trust-based routing approaches can gracefully mitigate insider packet drop attacks by building trusted paths to the destination. Moreover, they showed that trust-based routing improves the packet's successful delivery under insider packet drop attacks over routing algorithms that do not consider trust. Clearly, the effectiveness of these trust-based routing algorithms is based on their underlying trust models. A good trust model will help the routing algorithm to quickly and accurately identify inside packet drop attackers and find alternate routes to avoid them.

3.3. A Selective Forwarding-Based DoS Attack and Its Analysis

3.3.1. Motivation. the current trust mechanisms and trust-based routing cannot detect all known insider packet drop attacks. For instance, an intelligent attacker who can keep its trust value above the threshold value Θ_T will not be detected. More weakness can be found in the literature such as [12]. Our proposed selective forwarding-based DoS attack comes from the following simple observation.

To attack victims and avoid being identified, the attacker node A will have to disguise itself by forwarding packets for some nodes. When a node M sends only its own packets to the attacker A and uses its watchdog to monitor A, apparently A cannot drop all the packets without being detected. However, if M also forwards packets from other nodes to A, then A may be able to drop all the packets from one or multiple victim nodes.

For example, in the WSN shown in Figure 1 where all the nodes generate packets with the same frequency and send them to the BS, node 2 can pick node 10 as its victim and drops all the packets from node 10. Therefore the BS can never hear messages from node 10 and hence comes the name "denial of service" for this attack. However, if node 2 forwards all the packets from nodes 3, 4, 5, 9, and 15 to the BS, when Beta trust model is used with $\Theta_T = 0.70$, the monitoring node 3 will hear node 2 forwarding $5/6 \approx 83\%$ of the packets and fails to identify node 2 as an attacker because node 2's trust value will be approximately 0.83, higher than the threshold $\Theta_T = 0.70$.

It is not hard to see that once an attacker positions itself on the routing path of many nodes, it can select multiple victim nodes and launch the denial of service attack without being noticed. This can easily cause a lot of damage to the network and so we need to find countermeasures to defeat such attack.

3.3.2. Protocol of the Attack. Steps shown in Algorithm 1 define the protocol for an inside attacker A to launch the selective forwarding-based DoS attacks against multiple victim nodes.

On each received packet (step 3), the attacker A first determines the direct sender of the message (node M) and original source node S that generates the packet (steps 4 and 5). If A has received packets from S before (i.e., S is not a new source node), A will either drop or forward the packet based on whether S is a victim or not (steps 6 and 7). If S is a new source node, A will update the number of nodes whose packets are routed to A through M by n_M++ (step 9). When n_M reaches a predetermined value, A will be able to select a new victim to launch the DoS attack (steps 10–14). We called this attack *selective forwarding-based DoS* because the attacker can selectively choose the victims and drop all the packets from the victims to mislead the BS to consider that the victim nodes are either out of service or disconnected.

3.3.3. Analysis of the Attack. For an inside attacker to launch the selective forwarding-based DoS attack against the victim nodes, the attacker needs to (i) be able to tell whether a received packet is from the victim nodes, and (ii) ensure that, after dropping all the packets from the victim nodes, the attacker will not be detected by the monitoring nodes.

We first show that assumption (i) is valid. In a geographic routing employed WSN, the receiver of a packet can obtain the source node (the node that creates the packet)

```
1  for each node $M$ that forwards packets directly to $A$ and monitors $A$ with its watchdog, $k_M = 0$; $n_M = 0$;
2  while (both the network and node $A$ are on) {
3  on the reception of a packet {
4    identify the node $M$ that forwards the message;
5    identify the source node $S$ that generates the packet;
6    if $S$ is a victim node, drop the packet;
7    if $S$ is a non-victim source node, forward the packet;
8    if $S$ is a new source node {
9      $n_M$++;
10     if $n_M = V[k_M]$ {
11       pick a new victim source node;
12       $k_M$++;
13       if $S$ is the new victim node, drop the packet;
14       else forward the packet;
         }}}}
```

Algorithm 1: Steps for attacker A to launch selective forwarding-based DoS attacks against multiple victim nodes.

information from the packet because the receiver is a legitimate relay node that can access the packet's header where the source identification is stored [21, 25]. Even when the source node is protected by methods such as authorization, it is still possible for a malicious receiver to figure out the source node information by breaking the authorization mechanism or analyzing network traffics [26, 27].

Second, we will show that requirement (ii) can be satisfied. Because the inside attacker is a legitimate member of the WSN, it knows the trust model and the threshold value Θ_T used in the network. In a well-defined trust mechanism, when a node's packet drop rate increases, its trust value should not increase. A node will be considered as trustworthy if its trust value is above the threshold. Therefore, an attacker can evaluate its own trust value and drop a packet only when a drop will not bring its trust value below the threshold Θ_T. In our proposed protocol, the attacker selects a victim only when there are enough nonvictim source nodes to keep the attacker's trust value above Θ_T. This is guaranteed by the carefully determined array $V[\cdot]$ used in step 10 as we will explain next.

We define $V[j]$ as the minimum number of source nodes whose packets are routed to the attacker (A) through the same monitoring node (M) such that the attacker can drop packets from $j + 1$ of these nodes without being detected by M. That is, $V[0]$ is the minimum number of nodes for attacker A to cover/disguise the first victim; $V[1]$ is the minimum number of nodes for A to attack two victims.

In the beta trust model, if attacker A attacks $(j+1)$ victims among $V[j]$ nodes and forward the packets for the other $(V[j] - (j + 1))$ nodes, its trust value will be

$$\frac{V[j] - (j+1)}{V[j]} = 1 - \frac{j+1}{V[j]}. \tag{5}$$

To keep this trust value higher or equal to the trust threshold Θ_T, we can easily obtain the following:

$$V[j] = \left\lceil \frac{j+1}{1-\theta_T} \right\rceil. \tag{6}$$

Table 1: Values of $V[j]$ for three trust models.

$V[j]$	$V[0]$	$V[1]$	$V[2]$	$V[3]$	$V[4]$	$V[5]$
T_{Beta}	4	7	10	14	17	20
T_{Entr}	19	38	57	76	94	113
T^*_{Entr}	7	14	21	28	35	42

For the entropy trust models, there is no closed formula for $V[j]$. However, we can compute $V[j]$ numerically for any given Θ_T. Table 1 lists the values of $V[j]$ for the three different trust models where 0.70 is used as the trust threshold Θ_T.

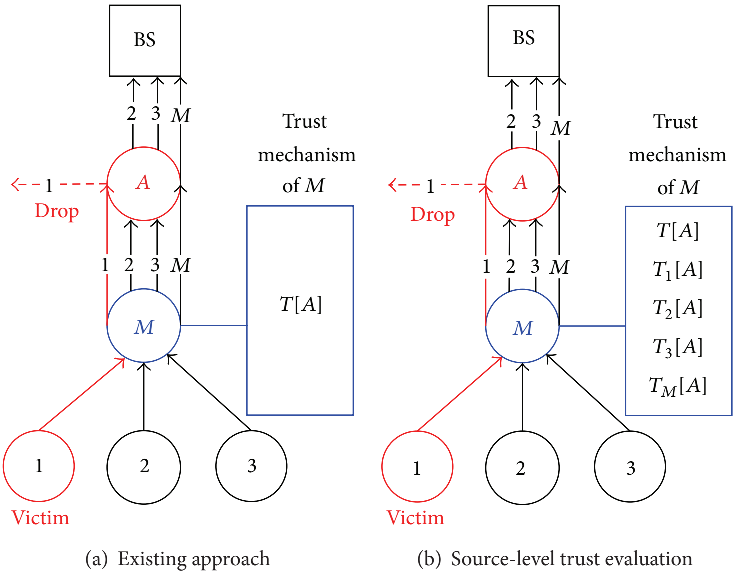
The small values of $V[0]$ indicate that the proposed selective forwarding-based DoS attack is a very serious threat. For the attacker (A) to launch the attack against a specific victim (V), it only requires the node (M) that forwards V's packets to A also forwards packets from 2 other nodes to A in the beta trust model ($V[0] = 4$: M, V, and 2 other nodes).

One can also see that $V[j]$ has a much larger value for the entropy trust models than the beta trust model. This is because earning a high trust value in entropy trust models (3) and (4) is much harder (i.e., a node must have very few packet drops) than earning a high trust value in the beta trust model (1).

4. The Proposed Defensive Mechanism

In this section, we propose and analyze our defensive mechanism, which is an enhancement of the beta and entropy trust mechanisms, against the above selective forwarding-based DoS attack. This defensive mechanism consists of two phrases: attacker detection and attacker-aware rerouting, which will be elaborated in Subsections 4.1 and 4.2 of this section, respectively. We analyze our approach and compare with existing methods in Subsection 4.3.

4.1. Source-Level Trust Evaluation and Attacker Detection. As depicted in Figure 2(a), in the existing trust mechanism [3, 9, 10, 14, 17, 22, 23, 28], a monitoring node M counts the



(a) Existing approach (b) Source-level trust evaluation

Figure 2: Existing trust evaluation approach [3, 9, 10, 14, 17, 22, 23, 28] and our proposed approach.

number of successes s and failures f that the next node A forwards packets from M. It then evaluates the trust value $T[A]$ of A based on s and f using the trust model adopted by the network. If $T[A] < \Theta_T$, M will consider A as an inside attacker. However, we have seen that this mechanism fails to detect intelligent attackers such as those launching the selective forwarding-based DoS attacks. For example, attacker A can drop all packets from node 1 but forwards packets from all other nonvictim nodes (in this case, nodes 2, 3, and M) to keep its trust value $T[A]$ high. When $T[A] \geq \Theta_T$, A's malicious attacking behavior will not be detected by M.

We can see that the current trust mechanism fails because the attacker can hide its malicious behavior behind its good behavior. As an attacker can identify the source node of a packet to launch the selective forwarding-based DoS attack, a monitoring node can also utilize the source node information to defend against such attack. This leads us to the following idea. If M uses separate counters to track not only A's overall packet forwarding behavior, but also how it delivers packets from each individual source node, then M will be able to tell whether A has launched the DoS attack against any node. This is shown in Figure 2(b) where M also evaluates A's trust value $T_i[A]$ for each source node i. We refer to this approach as *source-level trust evaluation*, and it can be easily integrated into the current 3-stage trust mechanism to improve its effectiveness of detecting inside attackers as follows.

Neighbor Behavior Monitoring. In addition to recording A's overall behavior s and f, for each packet that M overhears A is forwarding, M checks the source node information and updates a pair of separate counters, s_i and f_i, where i is the source node of the packet, to keep track the number of successes and failures for packets that A forwards from source node i, according to A's packet forwarding behavior to node i.

Trust Measurement. Based on the data collected in the first stage, M evaluates not only A's overall trust value $T[A]$ based on s and f, but also its source-level trust values $T_i[A]$ based on (s_i, f_i) to see how much M can trust A in forwarding packets from source node i. When the beta trust model is used, A's source-level trust value for source node i, $T_{\text{Beta},i}[A]$, can be calculated by using (1) as

$$T_{\text{Beta},i}[A] = \frac{s_i + 1}{s_i + f_i + 2}. \tag{7}$$

When the entropy trust model is used, A's source-level trust value for source node i, $T_{\text{Entr},i}[A]$, can be calculated by using (3) as

$$T_{\text{Entr},i}[A] = \begin{cases} 1 - H(p_i) & \text{for } 0.5 \leq p_i \leq 1; \\ H(p_i) - 1 & \text{for } 0 \leq p_i < 0.5, \end{cases} \tag{8}$$

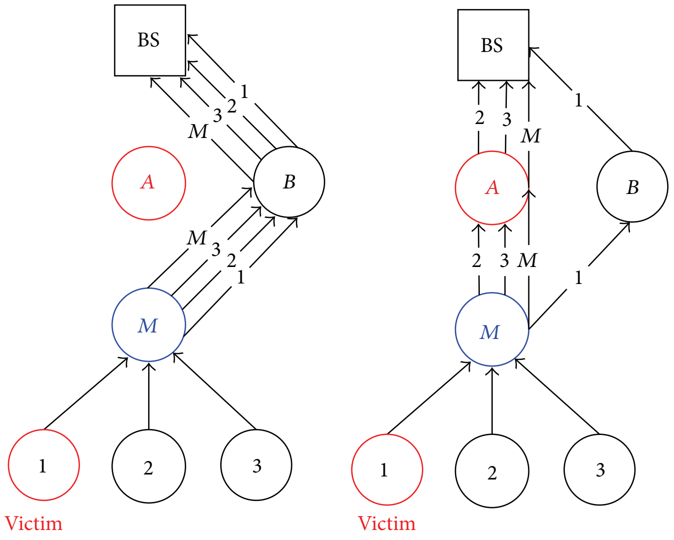
where $H(p_i) = -p_i \log_2 p_i - (1-p_i)\log_2(1-p_i)$ and $p_i = (s_i+1)/(s_i + f_i + 2)$. To have a nonnegative trust value between 0 and 1, we define

$$T^*_{\text{Entr},i}[A] = \frac{1 + T_{\text{Entr},i}[A]}{2}. \tag{9}$$

Detection. If any trust value $T_i[A]$ goes below the predetermined trust threshold Θ_T, M detects that A is a selective forwarding attacker against node i, the victim of such attack. When the overall trust value $T[A]$ of node A goes below the trust threshold Θ_T, A will be considered as an inside attacker just like the current trust mechanism will do.

Theorem 1. *The proposed source-level trust evaluation approach can successfully detect selective forwarding-based DoS attacks against any source node.*

Proof. By the definition of the selective forwarding-based DoS attack, if A launches attack against node i, it will behave



(a) Complete avoidance (CA)

(b) Selective avoidance (SA)

Figure 3: Two avoidance strategies to reroute the victim's packets to BS.

like a blackhole attacker and drop all packets originated from node i. Hence, after the attack is launched, s_i will remain unchanged and f_i will increase by one whenever a packet from node i is dropped by attacker A. When node i generates sufficient number of packets, the packet drop rate $\alpha = f_i/(s_i + f_i)$ will increase and can be arbitrarily close to 1. This means that A's trust value with respect to node i, $T_i[A]$, will approach to the minimum trust value, which will be way below the trust threshold Θ_T. So the monitoring node M will be able to identify this DoS attack and its victim.

Formally, let $n_i = s_i + f_i$ be the total packets generated by a victim node I; this theorem is based on the following fact:

$$\begin{aligned}\lim_{n_i \to \infty} T_i[A](\alpha) &= \lim_{n_i \to \infty} T_i[A]\left(\frac{f_i}{(s_i + f_i)}\right) \\ &= T_i[A](1) \ll \theta_T.\end{aligned} \tag{10}$$

Because all the (s_i, f_i) pairs are kept independently, the selective forwarding-based DoS attack against any other source nodes can also be detected, depending on how fast the victim nodes generate packets. □

Our approach requires the number of delivery successes and failures for packets from each source node. This will introduce storage overhead. Fortunately, such overhead is negligible. Even in the case when a node is receiving packets from 100 different source nodes and wants to track the status of the last 1 million packets from each node, the memory requirements will only be 0.25 KB (= $100 \times \log_2 2^{20}$ bits/8). This overhead is low for current sensors such as TelosB (10 KB RAM, 48 KB Flash, and 1 MB EEPROM) and Mica2/MicaZ (4 KB RAM, 128 KB Flash, and 512 KB EEPROM) [29].

4.2. Attacker-Aware Avoidance Routing Strategies. Once the attacker and a victim of the selective forwarding-based DoS attack are detected, approaches to reroute the victim's packet to the BS should be developed. In this section, we propose two attacker-aware rerouting algorithms, which we refer to as avoidance strategies.

When the value of a $T_i[A]$ becomes less than the network's trust threshold Θ_T, the monitoring node M will conclude that A is an inside attacker attacking node i. To avoid further damage that A may make to the network, M can use a *complete avoidance* (CA) strategy to reroute all the packets to another trustworthy neighbor node (such as B shown in Figure 3(a)). This ensures that all the packets received by M, not only those from node i, will avoid the attacker A. However, this strategy will increase the traffic on node B and may also introduce other routing overhead. For example, if node A was the best choice in an energy-efficient routing algorithm, rerouting all the packets to node B instead of A will cause increase in energy consumption. Furthermore, if A targets multiple victims, this strategy will help all of the victims to avoid the attacker A, but it can only identify the first victim. Finally, if M mistakenly claims the first victim, node A will be treated as an attacker. This will increase the false alarm rate in finding inside attackers.

In light of the fact that a selective forwarding-based DoS attacker (node A in this case) has targeted victims, the *selective avoidance* (SA) strategy will only reroute the discovered victim's packets to avoid the attacker A and keep the other packets running through node A (see Figure 3(b)). The monitoring node M will continue updating the trust values ($T_i[A]$) for all nodes except those discovered victims. So even when the attacker targets multiple victims, the SA strategy can discover all of them and help them avoid the attacker. This strategy will effectively solve CA's resource overhead problem. Its drawback is that it will take time for each of the victims to be identified, and the attacker can still drop packets from the victims and do damage to the network until all the victims are discovered.

Table 2: Comparison of the complete avoidance strategy (CA) and the selective avoidance strategy (SA).

	CA	SA
Reroute victim's packets	Yes	Yes
Reroute non-victim's packets	Yes	No
Time to reroute all victims' packets	Short	Long
Discover multiple victims	No	Yes
Probability of false alarm on attacker	Large	Small
Impact on the original routing solution	Large	Small

We summarize the features of the two proposed attacker-aware rerouting algorithms in Table 2.

4.3. Analysis of the Proposed Defensive Mechanism. The proposed defensive mechanism follows the 2-phase detection-avoidance framework. In the first phase, the source-level trust evaluation approach will detect victims of the selective forwarding-based DoS attack. In the second phase, the attacker-aware rerouting strategy will find a different path to deliver victim's packets to the BS.

4.3.1. Comparison with the Existing Trust Mechanisms. Our source-level trust evaluation is an enhancement of the existing trust-based mechanisms for inside attacker detection [10, 14]. The difference is that existing approaches do not consider the packet forwarding behavior of the receiving node (the node being monitored) for each individual source node. Therefore, it can detect whether the node is an inside packet drop attacker, but it will fail to detect the proposed selective forwarding-based DoS attack. In our proposed method, the monitoring node will evaluate the trust value with respect to each source node. As stated in Theorem 1, this enhancement enables us to identify not only the attacker, but also all the victims. The cost of our approach, compared with existing mechanisms, is the storage requirement to keep the delivery information for each source node, which we have analyzed after the proof of Theorem 1.

Now we compare the false alarm rate (FAR) of our approach with existing mechanisms. FAR measures how likely a good node will be tagged as an inside attacker. Let FAR, $\mathrm{FAR}_{\mathrm{CA}}$, and $\mathrm{FAR}_{\mathrm{SA}}$ be the FAR of the existing detection approach, our approach with CA, and our approach with SA, respectively. We have the following.

Theorem 2. $\mathrm{FAR}_{\mathrm{CA}} \geq \mathrm{FAR} \geq \mathrm{FAR}_{\mathrm{SA}}$.

Proof. Recall that the trust value $T[A]$ in the existing trust mechanism is defined based on the packet drop rate, which is the ratio of the total failures (f) over the total number of packets $(s+f)$. A false alarm occurs when a good node's trust value $T[A]$ becomes smaller than the trust threshold Θ_T. In our approach, the monitoring node M also updates $T_i[A]$, the trust value with respect to source node i, which is determined by the drop rate of packets from node i or the pair of (s_i, f_i).

When we use CA strategy in the second phase, the first detected victim node j is the one that has the largest packet drop rate that results in the smallest $T_i[A]$ among all the node i's that send their packets to A through the same monitoring node M. That is, $T_j[A] = \min\{T_i[A]\}$. For the same set of node i's, we have $s = \sum s_i$ and $f = \sum f_i$. Clearly, $T[A] \geq T_j[A]$. Therefore, when the existing detection mechanism claims (regardless of the correctness of the claim) node A as an attacker (i.e., $T[A] < \Theta_T$), our approach should have already identified the first victim j of A's DoS attack because

$$T_j[A] \leq T[A] < \Theta_T. \tag{11}$$

However, when our approach claims an attacker, $T_j[A] < \Theta_T$, it is not necessarily true that $T[A] < \Theta_T$. A false alarm is an incorrect claim. So $\mathrm{FAR}_{\mathrm{CA}} \geq \mathrm{FAR}$.

On the other hand, when SA strategy is applied, our approach will identify the DoS victims one by one and reroute the packets from these victims to nodes other than the attacker A. Note that victim nodes always have large packet drop rate; when their packets are rerouted, the trust value of A evaluated by our approach will be higher than that in the existing approach. This is because the existing approach will count the (s_i, f_i) pairs from these victims in $s = \sum s_i$, $f = \sum f_i$ in the calculation of $T[A]$. So when the same Θ_T is used, $T[A] < \Theta_T$ will always first happen in the existing approach before it happens in our approach with SA strategy, that is, $\mathrm{FAR} \geq \mathrm{FAR}_{\mathrm{SA}}$. □

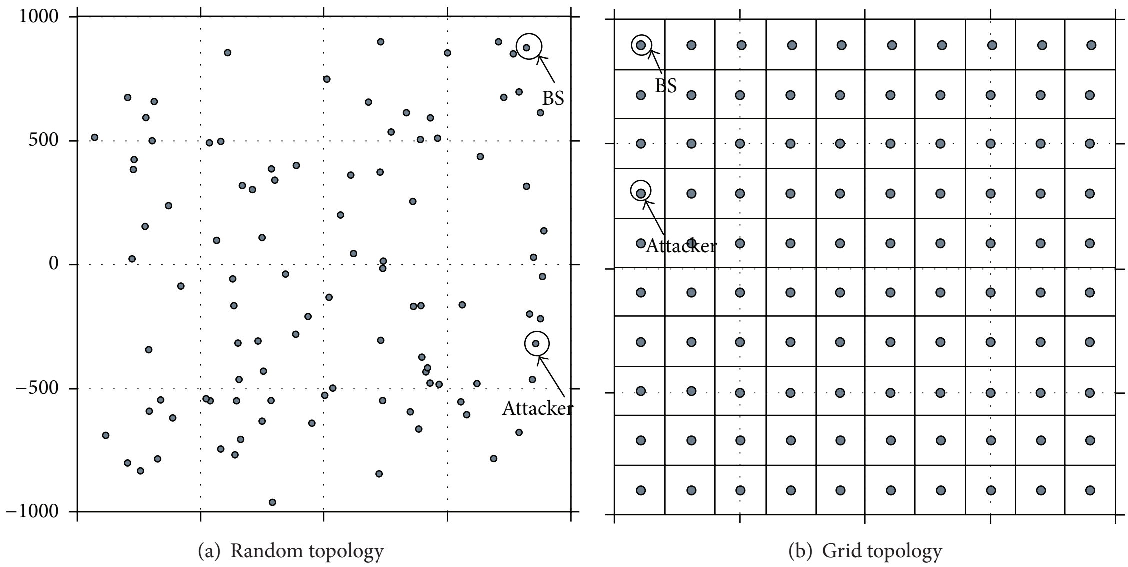
4.3.2. Comparison with the Avoidance Approaches. As we have mentioned in the introduction, the idea behind current avoidance approaches is to send packets from multiple disjoint paths in order to avoid inside packet drop attackers [2, 13, 18, 19]. These approaches cannot and are not intended to detect the attackers. We have also discussed in Section 2 that the overhead of such avoidance approaches can be prohibitively high. For example, when each packet is sent through multiple different paths, the transmission energy, the network traffic, and collision will all increase dramatically.

Despite the same name, the avoidance strategy in the second phase of our defensive mechanism is conceptually different from the above avoidance approaches. In our approach, the avoidance strategy is applied after both the victims and the attacker in the insider packet drop attack have been identified. Therefore we can efficiently find a path that does not involve the attacker to deliver victim's packets to the BS. Although the new path may not be as good as the initial path (where the attacker sits on) in terms of energy, delay, or channel quality, neither CA nor SA uses multiple paths. Hence, the large overhead problem in the conventional avoidance approaches does not exist in our defensive mechanism.

5. Simulation and Results Analysis

5.1. Simulation Goals, Setups, and Evaluation Metrics. There are two main goals of the simulation: validating that the current trust mechanisms fail to detect the proposed DoS attack and evaluating the performance of our defensive approach.

The parameters in Table 3 are used in our simulations. We conduct simulations with the commercial network simulator



(a) Random topology

(b) Grid topology

Figure 4: Two WSN topologies in our simulations.

Table 3: OPNET simulation setup parameters for the validation of our detection mechanisms.

Parameters	Setting
General	
Terrain dimension	2 km × 2 km
Number of nodes	100
Topology	Random/grid
Max. simulation time	30 mins for single attacker; 40 mins for multiple attackers
Base routing algorithm	GRP
Max. retransmissions	7 (OPNET default)
Data packet generation	
Start time–stop time	100 seconds–end of simulation
Destination	Base station
Packet arrival interval	Every 10 second
Packet size	1,024 bits
Trust model	
Type	Beta/entropy
Initial trust value	0.99
Trust threshold (Θ_T)	0.7
Attack model	
Number of attackers	Single attacker/multiple attackers (=2)
Attack type	Selective forwarding-based DoS attack

OPNET Wireless Modeler v.17.1. 100 sensors are deployed in a 2 km × 2 km area randomly in one setting (Figure 4(a)) and in a 10 × 10 grid in another setting (Figure 4(b)). Each node except the BS generates packets randomly in each 10-second period. The packets are sent to the BS. We use some of the default settings in OPNET such as 1024 bit data packet and geographic routing protocol (GRP) with a maximum of 7 retransmissions before a packet is dropped. We set each node's initial trust value to be 0.99. We consider the cases of both single and multiple selective forwarding-based DoS attackers. The simulation time is set to be 30 minutes in the case of single attacker and 40 minutes for multiple attackers. We simulate the attacker(s) launch the proposed selective forwarding-based DoS attack to various numbers of victims. Both beta and entropy trust models (defined in (1) and (4)) as well as our enhanced trust mechanism (defined in (7) and (9)) with two avoidance strategies (CA and SA) are implemented in the OPNET Modeler for comparison purposes.

The main performance evaluation metrics are as follows.

(1) *Avoidance completion time (ACT)*: this is the time when all the victims have been rerouted to avoid the attacker.

(2) *False alarm rate (FAR)*: as discussed in the previous section, this is the probability that a good node is being considered as a selective forwarding-based DoS attacker.

(3) *Energy per packet (EPP)*: this is the average energy consumption to deliver a data packet, regardless of whether the packet reaches the BS or not. EPP is obtained by the total energy consumed for data packet transmissions divided by the total number of data packets generated by all source nodes.

5.2. Simulation Results and Analysis of Single Attacker. ACT is the most important metric as it indicates the ability of each approach in identifying the attacker and rerouting the victim's packets.

The simulation results on ACT in Table 4 reveal the following.

Beta or Entropy Trust Models Alone Fail to Detect the Attacker. In the grid topology, there are 21 source nodes that send

Table 4: Avoidance completion time (in seconds) for the Beta and Entropy trust model without any avoidance strategy (Pure), with the complete avoidance strategy (CA), and with the selective avoidance strategy (SA) when the attacker targets J victim nodes.

J	Beta trust model			Entropy trust model		
	Pure	CA	SA	Pure	CA	SA
			Grid topology			
1	Fail	542.5	542.5	Fail	269.0	269.0
2	Fail	539.0	551.5	Fail	266.0	279.5
3	Fail	538.5	552.0	Fail	268.0	278.5
4	Fail	538.0	553.5	255.5	255.5	255.5
5	Fail	543.0	562.5	184.5	184.5	184.5
6	Fail	541.5	552.5	152.5	152.5	152.5
7	581.0	541.5	561.0	151.5	151.5	151.5
			Random topology			
1	Fail	548.3	548.3	Fail	277.2	277.2
2	Fail	546.9	604.6	Fail	289.0	300.9
3	Fail	553.2	644.8	360.9	278.6	329.7
4	Fail	559.0	602.0	204.7	204.7	204.7
5	802.5	549.5	591.3	162.4	162.4	162.4
6	352.6	352.6	352.6	153.2	153.2	153.2

Table 5: False alarm rate in the random topology.

J	Beta trust model			Entropy trust model		
	Pure	CA	SA	Pure	CA	SA
1	0.010	0.036	0.010	0.048	0.069	0.048
2	0.010	0.034	0.010	0.045	0.067	0.044
3	0.011	0.033	0.010	0.065	0.065	0.046
4	0.011	0.031	0.010	0.064	0.064	0.064
5	0.031	0.035	0.010	0.061	0.061	0.061
6	0.028	0.028	0.028	0.061	0.061	0.061

Table 6: Energy per packet (mJ) for the Beta and Entropy trust model without any avoidance strategy (Pure), with the complete avoidance strategy (CA), and with the selective avoidance strategy (SA) when the attacker targets J victim nodes.

J	Beta trust model			Entropy trust model		
	Pure	Overhead (%)		Pure	Overhead (%)	
		CA	SA		CA	SA
			Grid topology			
1	37.53	3.65	0.40	37.53	4.42	0.51
2	37.42	3.87	0.77	37.42	4.73	0.99
3	37.27	4.19	1.23	37.27	5.12	1.50
4	37.14	4.52	1.67	39.19	0	0
5	37.03	4.70	2.05	39.25	0	0
6	36.89	5.04	2.49	39.32	0	0
7	38.65	0.13	−2.07	39.3	0	0
			Random topology			
1	78.14	1.10	0.44	78.40	1.38	0.54
2	77.56	1.55	0.98	77.74	2.14	1.40
3	76.96	2.27	1.68	79.23	0.06	−0.67
4	76.42	3.01	2.30	79.55	0	0
5	78.08	0.54	0.09	79.70	0	0
6	79.01	0	0	79.51	0	0

packets to the monitoring node and then to the attacker. From Table 1, when the selective forwarding-based DoS attacker targets 6 victims or less, the beta trust model will not detect it; when it targets 3 victims or less, the entropy trust model cannot detect it. The results in Table 4 confirm this. This is also true for the random topology where the monitoring node forwards packets from 16 source nodes (including itself) to the attacker.

Our Defensive Mechanisms Successfully Detect the Victims. Even when the attacker targets only one victim (the case of $J = 1$), our defensive mechanism can help both the beta trust model and the entropy trust model to identify the victim node. The entropy trust model is quicker because a dropped packet will cause more reduction in the trust value in the entropy trust model. It also takes more time for the SA strategy because it finds victims one by one.

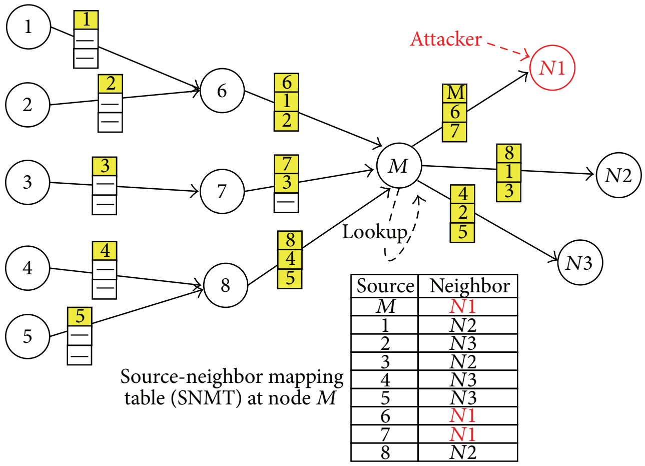
Optimality of the Proposed Selective Forwarding-Based DoS Attack. We already discussed in the first item that our proposed selective forwarding-based DoS attack cannot be detected by the current trust model. Table 4 also shows that if the attacker becomes aggressive and targets more victims than the $V[j]$ values in Table 1 allow, then they will be detected by both the beta trust model and the entropy trust model.

FAR measures the likelihood an approach will mistakenly treat an honest node as attacker. In the grid topology, there are very few collisions and there is no false alarm. The FAR values for different approaches in the random topology are shown in Table 5. This result confirms the claim of $\mathrm{FAR}_{\mathrm{CA}} \geq \mathrm{FAR} \geq \mathrm{FAR}_{\mathrm{SA}}$ we made in Theorem 2.

Finally, we report EPP. From Table 6, we can see that our proposed enhancement incurs very little energy overhead. In the avoidance approach where multiple paths are used, for a single attacker, two disjoint paths will guarantee the successful avoidance of the attacker. However, the energy consumption will be doubled. From energy perspective, our approach is much better than the current avoidance approach.

In a couple of cases, when SA strategy is used, there is actually a small amount of energy savings. This is possible because the original geographical routing protocol does not guarantee energy efficiency. Moreover, as we have analyzed, SA strategy uses less energy than CA strategy because in SA strategy, only packets from detected victims will be rerouted.

5.3. Simulation of Multiple Attackers. For simplicity, we report the case of two attackers. When the two attackers are far away from each other, launching attacks to victim nodes independently, the result for each attack is almost identical to the single attacker case. Here we discuss the more interesting case when the two attackers are physically close to each other, for example, when the node to the right of the attacker in



Source	Neighbor
M	N1
1	N2
2	N3
3	N2
4	N3
5	N3
6	N1
7	N1
8	N2

Figure 5: Our prevention routing algorithm against a selective forwarding-based DoS attacker ($N1$) when the beta trust model is used with $\Theta_T = 0.7$.

Table 7: Avoidance completion time (seconds) in the case of multiple attackers in the grid topology.

J	Beta trust model			Entropy trust model		
	Pure	CA	SA	Pure	CA	SA
1	Fail	1,896.6	1,895.3	Fail	805.3	807.3
2	Fail	1,876.6	1,447.3	Fail	780.6	632.0
3	Fail	1,862.6	1,306.6	Fail	772.6	574.6
4	Fail	1,861.3	1,230.0	8,750	742.6	624.6
5	Fail	1,869.3	1,189.3	617.3	563.3	475.3
6	Fail	1,866.6	1,158.0	372.0	372.0	372.0
7	Fail	1,864.0	1,137.3	298.0	298.0	298.0
8	Fail	1,374.0	999.3	248.6	248.6	248.6
9	1,438.3	1,092.6	835.3	215.3	215.3	215.3
10	740.0	740.0	740.0	196.6	196.6	196.6

Figure 4(b) is also an attacker and they both target the same set of victims.

As one can imagine, when a victim node is identified, either the CA or the SA strategy will try to reroute packets to avoid the attacker. However, because the attacker's neighbor is also an attacker, if the monitoring node happens to choose the second attacker to forward packets to, both ACT and EPP will increase. In particular, the ACT will be around doubled because it will take about the same amount of time for the monitoring node to recognize the second attacker and reroute again. We now study the simulation results below.

First, we see that the two attackers together can target more victims without being detected. For example, in Table 4, we know that a single attacker will be detected by the beta trust model if it attempts to attack 7 or more victims. However, Table 7 shows that the beta trust model can find the two attackers only when they are trying to attack 9 or more victims, which apparently indicates the improvement of attacking power.

Second, we see that the ACT is about tripled, instead of doubled, of the ACT in the single attack model. This is a little unexpected. However, the topology of the network and the position of the attackers are the main reason for this. In our case, when the monitoring node finds the second attacker, it will reroute the packets to a new node. The new node happens to forward the packets to the second attacker again; thus, it will take again time for the new node to identify the second attacker. This results in the ACT in the 2-adjacent attackers case is about three times of the ACT for single attacker.

6. Prevention Routing Algorithm

6.1. Motivation and Key Idea. As we have discussed earlier, when an inside attacker relays packets for many sensor nodes in the network, it can pick one or more victims to launch the selective forwarding-based DoS attack. This is because it can hide its malicious behavior by forwarding packets from other nodes and maintaining a high trust value. If an attacker is on the routing path of only one or two nodes and it attacks a victim, the chance that the attacker will be detected quickly is high. In such situation, the attacker may not take the risk to launch any attack. Based on this observation, we propose a prevention routing algorithm where an attacker has to choose between "not attacking" and "attacking and being caught." This is complementary to the detection and avoidance approach we described earlier. They can be used together as a more effective defensive mechanism.

The key idea of our prevention method is to limit the number of source nodes (N_{SMAX}) from which a node receives packets through the same monitoring node. As discussed in Section 3, if the attacker receives data packets from at least $V[0]$ source nodes from a monitoring node, it can launch the selective forwarding-based DoS attack against one of the source nodes without being detected by the monitoring node. Therefore, in our prevention method, we require each monitoring node forwards packets from at most $V[0] - 1$ source nodes; that is, $N_{\text{SMAX}} = V[0] - 1$. This will prevent the attacker from launching the selective forwarding-based DoS attack. If the attacker still launches the attack, it will be detected by the monitoring node.

Figure 5 shows how our prevention method successfully defends against a selective forwarding-based DoS attacker $N1$. Consider that the beta trust model with the trust

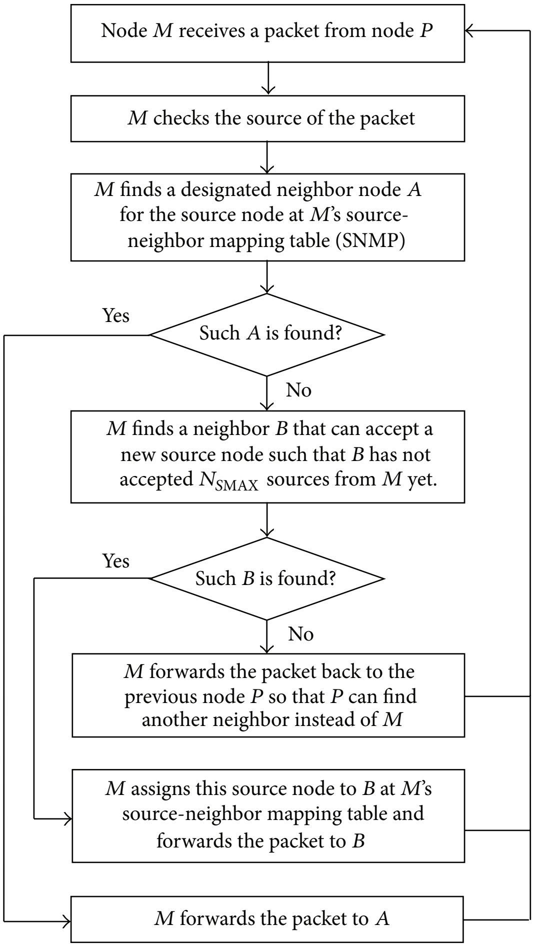

Node M receives a packet from node P
M checks the source of the packet
M finds a designated neighbor node A for the source node at M's source-neighbor mapping table (SNMP)
Such A is found?
Yes
No
M finds a neighbor B that can accept a new source node such that B has not accepted N_{SMAX} sources from M yet.
Such B is found?
Yes
No
M forwards the packet back to the previous node P so that P can find another neighbor instead of M
M assigns this source node to B at M's source-neighbor mapping table and forwards the packet to B
M forwards the packet to A

Figure 6: The flow chart of a trust-based routing algorithm with our prevention method to prevent the selective forwarding-based DoS attack.

threshold $\Theta_T = 0.7$ is used in a WSN. Node M receives packets from 8 nodes (nodes from 1 to 8). Although node $N1$ is M's best choice to relay the packets, our prevention routing algorithm will limit M to forward packets from only 3 nodes (nodes M, 6, and 7 in this case) to $N1$. Packets from the other 6 nodes will be forwarded to nodes $N1$ and $N2$, three each. That is, $N_{\mathrm{SMAX}} = 3$.

We know that $V[0] = 4$ in this case from Table 1. As a result, the attacker $N1$ cannot launch the selective forwarding-based DoS attack against any of the three source nodes (M, 6, and 7) without being detected by M. If $N1$ starts attacking any of the three source nodes, $N1$'s trust value evaluated by M will be 0.67 (=2/3) and thus $N1$ will be caught by M because $N1$'s trust value is less than Θ_T (=0.7).

6.2. Proposed Prevention Routing Algorithm. Our prevention method can be easily integrated into any existing trust-based routing algorithm. Figure 6 shows the flow chart of a trust-based routing algorithm with our prevention method. Each time node M wants to forward a data packet toward the BS (regardless of its own packet or packets it receives from other nodes), M first checks the source node of the data packet and then finds a neighbor node A at M's *source-neighbor mapping table* (SNMT). SNMT is a look-up table that tells M which of M's neighbors will receive a certain source node's data packet to forward the packet toward the BS. If such node A is found at the SNMT for the source node, M will forward the packet to A. Otherwise, M will find a new neighbor node B such that the number of source nodes assigned to B is less than N_{SMAX}. If there are multiple neighbors satisfying such condition, the next hop selection algorithm of a base routing algorithm such as GPSR will choose the best one among them. If such node B is found, M registers B to its SNMT for the source node and then forwards the data packet to B. If M cannot find any neighbor satisfying such condition, M forwards the source's data packet back to the previous node P so that P can find other neighbor instead of M.

We explain how a relay node M assigns source nodes to its neighbor nodes as shown in Figure 5. Assuming that every source's data packet is equally important, we use the (First Come First Serve) FCFS manner for this source-neighbor assignment process. For example, in Figure 5, assume that M received the first data packets of its eight source nodes in the following order: M, 6, 7, 8, 1, 3, 4, 2, and 5. Then, M assigns firstly arrived three source nodes (M, 6, 7) to its best neighbor $N1$ chosen by its base routing algorithm. The next three source nodes (8, 1, and 3) and the remaining three source nodes (4, 2, and 5) are assigned to M's next best neighbors

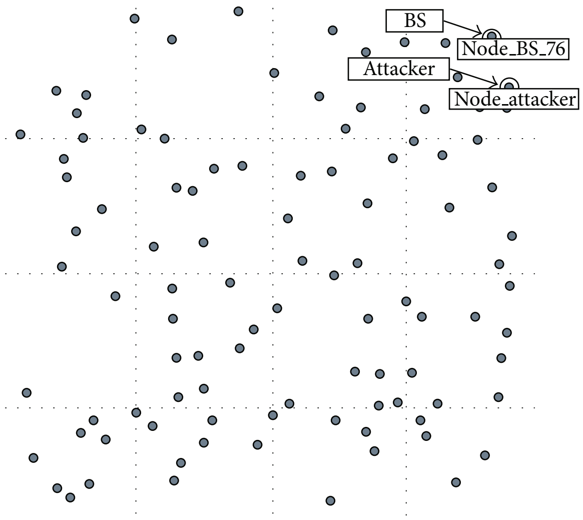


Figure 7: A WSN topology in our simulations. One hundred sensors are deployed in a 2 km × 2 km area randomly.

$N2$ and $N3$, respectively. Each source-neighbor pair is stored in M's SNMT. Whenever M receives a data packet, M will forward the packet to its designated neighbor associated with the source of the packet by using M's SNMT.

6.3. Simulation Setups and Preliminary Simulation Results. We use the simulation parameters described in Table 8. 100 sensors are deployed in a 2 km × 2 km area randomly as shown in Figure 7. Each node except the BS generates packets randomly in each 10-second period. The packets are sent to the BS. We use some of the default settings in OPNET such as 1024 bit data packet and geographic routing protocol (GRP) with a maximum of 7 retransmissions. We set each node's initial trust value to be 0.99. We choose one node near the BS as the selective forwarding DoS attacker. The simulation time is set to be 60 minutes. The attacker targets various numbers of victims. We implement two trust-based routing algorithms: trust-based GRP based on the beta trust model (Beta GRP) and our prevention routing algorithm combining the Beta GRP and our prevention method (Beta GRP-P). For our prevention method, N_{SMAX} is set to be 3 because the beta trust model with $\Theta_T = 0.7$ is used in simulations.

In addition to the three performance metrics (ACT, FAR, and EPP) used in Section 5, we use the following two performance metrics.

(1) Number of source nodes whose data packets route to the attacker through the same monitoring node (N_{S_i}): by using this metric, as we discussed in Section 3, we can get the theoretical maximum number of victims (N_{VMAX}) which the attacker can stealthy target without being noticed by the BS.

(2) Packet delivery rate (PDR): this is the probability that a data packet is delivered to the BS. PDR is obtained by the total number of data packets delivered to the BS divided by the total number of data packets generated by all source nodes.

Table 8: OPNET simulation setup parameters for the validation of the our prevention routing algorithm.

Parameters	Setting
General	
Terrain dimension	2 km × 2 km
Number of nodes	100
Topology	Random
Max. simulation time	60 mins (3,600 seconds)
Base routing algorithm	GRP
Max. retransmissions	7 (OPNET default)
Data packet generation	
Start time–stop time	100 seconds–end of simulation
Destination	Base station
Packet arrival interval	Every 10 second
Packet size	1,024 bits
Trust model	
Type	Beta trust model
Initial trust value	0.99
Trust threshold (Θ_T)	0.7
Attack model	
Number of attackers	Single attacker
Attack type	Selective forwarding-based DoS attack

We first show how many source nodes' data packets can route through the inside attacker (located near the BS) in the simulation network topology. To see routing paths from source nodes to the BS via the attacker, we simulate the attacker forwarding packets normally toward the BS without attacking any source (attack off). Figures 8 and 9 show source nodes whose data packets route through the attacker and their routing paths to the BS when the beta GRP and our prevention routing algorithm (Beta GRP-P) are used, respectively. We can see that when beta GRP is used, the attacker receives data packets from many more source nodes as compared to our approach used.

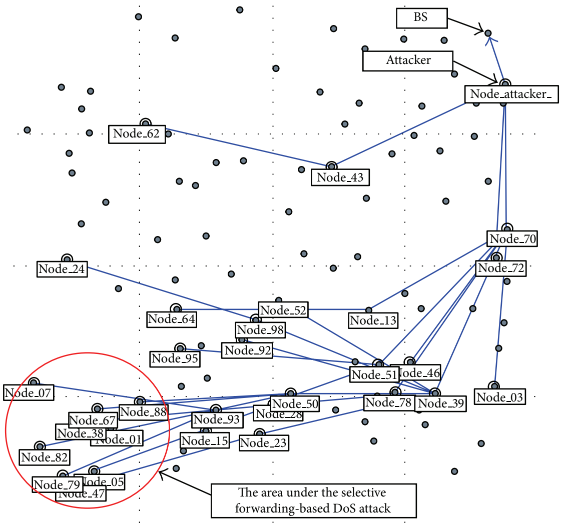


Figure 8: 30 potential victim source nodes and their routing paths to the BS when Beta GRP is used.

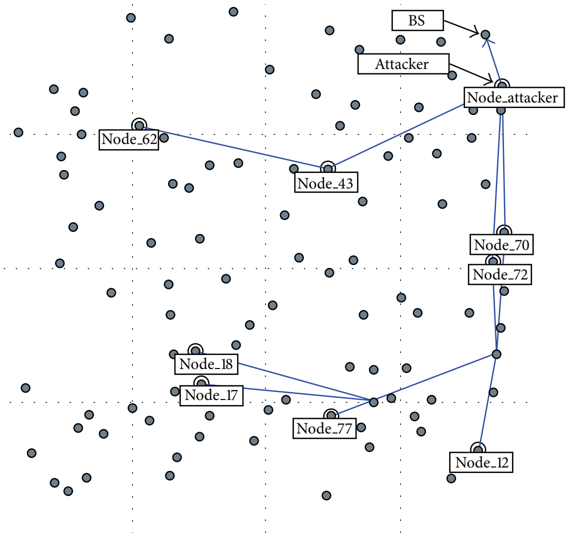


Figure 9: 8 potential victim source nodes and their routing paths to the BS when our prevention routing algorithm (Beta GRP-P) is used.

Specifically, Table 9 shows N_S and N_{VMAX} of the beta GRP and our approach (Beta GRP-P). For example, as the third row in bold shows, when the beta GRP is used, the attacker receives 27 source nodes' data packets from node 70. In this case, the attacker can drop up to 8 nodes' data packets completely without being detected by node 70's beta trust model theoretically ($N_{\mathrm{VMAX}} = 8$). Meanwhile, when our approach is used, the attacker receives at most 3 source nodes from node 70 or 72. Consequently, the attacker cannot successfully launch the DoS attack against any source node without being detected by node 70 or 72.

Table 9: The number of source nodes whose data packets route through the attacker (N_S) and the maximum number of victim source nodes (N_{VMAX}).

Monitoring node	Beta GRP		Our beta GRP-P	
	N_S	N_{VMAX}	N_S	N_{VMAX}
43	2	0	2	0
70	**27**	**8**	**3**	**0**
72	1	0	3	0
Total	30	8	8	0

Second, we examine ACT that indicates the ability of each approach in identifying the attacker and rerouting the victim's packets. We simulate the attacker launching the selective forwarding-based DoS attack by increasing the number of victims (J). We assume that the attacker intentionally targets source nodes from node 70 because it can have the largest number of victim source nodes. The simulation results on ACT in Table 10 reveal the following.

(1) *Beta trust model alone fails to detect the attacker*. As shown in Table 10, the attacker can attack up to 8

Table 10: Avoidance completion time (in seconds) comparison when the attacker targets J victim nodes. In our prevention routing, the attacker can target at most 3 nodes.

J	Beta GRP	Our beta GRP-P
1	Fail	1,296
2	Fail	360
3	Fail	216
4	Fail	N/A
5	Fail	N/A
6	Fail	N/A
7	Fail	N/A
8	Fail	N/A
9	432	N/A

sources without being caught by node 70's beta trust model. As shown in Figure 8, the entire area monitored by the 8 victims (circled area) can be influenced by the DoS attack, and thus outside intruders can stay in or move around the area stealthily.

(2) *Our approach successfully defends against the attacker.* The proposed outsider-insider colluding attack is not effective when our prevention approach is used because the number of victims is very small. That is, since the attacker cannot target more than 3 source nodes when our approach is used, the victim area is significantly reduced compared to when the Beta GRP is used. As a result, outside intruders' movement will be limited by the small area monitored by victim nodes. In addition, if the attacker insists to attack any victim (the case of $J = 1$), the attacker will be detected by 1,296 seconds. This ACT can be reduced to around 540 seconds when our detection scheme is used together (see Table 4).

Third, FAR measures the likelihood an approach will mistakenly treat an honest node as attacker. Both approaches have almost similar FARs that range from 0.03 to 0.04. Thus, we consider that our approach does not increase FAR compared with the Beta GRP.

Finally, we report PDR and EPP. We show results when the number of victims (J) is less than 4 for comparison purposes. As shown in Table 11, our approach has a higher packet delivery performance than the beta GRP. This is because our approach can detect and avoid the attacker while the Beta GRP cannot defend against the attacker. In addition, from energy perspective, we can see that our approach is better than the Beta GRP.

7. Conclusions

In this paper, we first present a simple selective forwarding-based DoS attack and show that two representative trust mechanisms (namely, the beta trust model and the entropy trust model) fail to detect such attack. We also show the potential damage this attack could cause to the network. Second, we propose a source-level trust evaluation scheme to enhance the beta and entropy trust mechanisms to effectively detect the selective forwarding-based DoS attack. In addition, we propose two avoidance strategies to reroute the victim's packets so they can reach the BS and validate our claims and evaluate the performance of our detection and avoidance mechanisms with extensive OPNET simulations.

Table 11: Packet deliver rate (PDR) and energy per packet (mJ); J: number of victim source nodes.

J	Beta GRP		Our beta GRP-P	
	PDR	EPP	PDR	EPP
1	0.910	92.37	0.946	87.63
2	0.901	92.29	0.957	85.10
3	0.893	92.10	0.945	86.15

Finally, we introduce a prevention-routing algorithm to proactively prevent the selective forwarding-based DoS attack as a complementary defensive mechanism to our detection and avoidance methods and provide preliminary results to evaluate its performance.

There are also several directions for future work. First, how to further reduce ACT to minimize the attacker's damage to the network. Second, our preliminary results on network with lossy network show fairly large FAR. How to improve the accuracy of the proposed approach in such network is still a challenge. Finally, after the inside attackers become aware of our defensive mechanism, how they can respond to the challenge and launch more sophisticated attacks.

Acknowledgments

This material is based upon work supported in part by the Air Force Office of Scientific Research (AFOSR/RSL) under Award no. #FA95501010140 and a University Partnership with the Laboratory of Telecommunications Sciences, Contract no. H9823013D00560002. The authors would like to thank OPNET Technologies, Inc., for providing us with OPNET Wireless Modeler to validate our approaches. There is no conflict of interests.

References

[1] S. Djahel, F. Naït-Abdesselam, and Z. Zhang, "Mitigating packet dropping problem in mobile ad hoc networks: proposals and challenges," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 4, pp. 658–672, 2011.

[2] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[3] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 255–265, August 2000.

[4] T. H. Hai and E. N. Huh, "Detecting selective forwarding attacks in wireless sensor networks using two-hops neighbor knowledge," in *Proceedings of the 7th IEEE International Symposium on Networking Computing and Applications (NCA '08)*, pp. 325–331, July 2008.

[5] I. Khalil, S. Bagchi, C. N. Rotaru, and N. B. Shroff, "UnMask: utilizing neighbor monitoring for attack mitigation in multihop

wireless sensor networks," *Ad Hoc Networks*, vol. 8, no. 2, pp. 148–164, 2010.

[6] X. S. Wang, Y. Z. Zhan, S. M. Xiong, and L. M. Wang, "Lightweight defense scheme against selective forwarding attacks in wireless sensor networks," in *Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC '09)*, pp. 226–232, October 2009.

[7] B. Yu and B. Xiao, "Detecting selective forwarding attacks in wireless sensor networks," in *Proceedings of the 20th Parallel and Distributed Processing Symposium (IPDPS '06)*, April 2006.

[8] Y. Yu, K. Li, W. Zhou, and P. Li, "Trust mechanisms in wireless sensor networks: attack analysis and countermeasures," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 867–880, 2012.

[9] A. A. Pirzada and C. McDonald, "Trusted greedy perimeter stateless routing," in *Proceedings of the 15th IEEE International Conference on Networks (ICON '07)*, pp. 206–211, November 2007.

[10] Y. L. Sun, W. Yu, and Z. Han, "Information theoretic framework of trust modeling and evaluation for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 305–315, 2006.

[11] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 3, article 15, 2008.

[12] Y. Cho, G. Qu, and Y. Wu, "Insider threats against trust mechanism with watchdog and defending approaches in wireless sensor networks," in *Proceedings of the IEEE Symposium Security and Privacy Workshops (SPW '12)*, pp. 134–141, 2012.

[13] H. M. Sun, C. M. Chen, and Y. C. Hsiao, "An efficient countermeasure to the selective forwarding attack in wireless sensor networks," in *Proceedings of the IEEE Region 10 Conference (TENCON '07)*, pp. 1–4, November 2007.

[14] A. Josang and R. Ismail, "The beta reputation system," in *Proceedings of the the 15th Bled Electronic Commerce Conference*, June 2002.

[15] X. Su and R. V. Boppana, "On mitigating in-band wormhole attacks in mobile ad hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, pp. 1136–1141, June 2007.

[16] Q. Yan, M. Li, T. Jiang, W. Lou, and Y. T. Hou, "Vulnerability and protection for distributed consensus-based spectrum sensing in cognitive radio networks," in *Proceedings of the IEEE h International Conference on Computer Communications (INFOCOM '12)*, pp. 900–908, 2012.

[17] T. Zahariadis, H. Leligou, P. Karkazis et al., "Design and implementation of a trust-aware routing protocol for large WSNs," *International Journal of Network Security & Its Applications*, vol. 2, no. 3, pp. 52–68, 2010.

[18] B. Deb, S. Bhatnagar, and B. Nath, "ReInForM: reliable information forwarding using multiple paths in sensor networks," in *Proceedings of the IEEE Local Computer Networks (LCN '03)*, pp. 406–415, October 2003.

[19] J. Deng, R. Han, and S. Mishra, "Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks," in *Proceedings of the International Conference on Dependable Systems and Networks*, pp. 637–646, July 2004.

[20] "OPNET modeler wireless suite," 2012, http://www.opnet.com/solutions/network_rd/modeler_wireless.html.

[21] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, August 2000.

[22] S. Tanachaiwiwat, P. Dave, R. Bhindwale, and A. Helmy, "Location-centric isolation of misbehavior and trust routing in energy-constrained sensor networks," in *Proceedings of the 23rd IEEE International Performance, Computing, and Communications Conference (IPCCC '04)*, pp. 463–469, April 2004.

[23] N. Bhalaji, S. Banerjee, and A. Shanmugam, "A novel routing technique against packet dropping attack in adhoc networks," *Journal of Computer Science*, vol. 4, no. 7, pp. 538–544, 2008.

[24] P. Poonam, K. Garg, and M. Misra, "Trust based multi path DSR protocol," in *Proceedings of the 5th International Conference on Availability, Reliability, and Security (ARES '10)*, pp. 204–209, February 2010.

[25] I. Khalil and S. Bagchi, "Stealthy attacks in wireless ad hoc networks: detection and countermeasure," *IEEE Transactions on Mobile Computing*, vol. 10, no. 8, pp. 1096–1112, 2011.

[26] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, pp. 599–608, June 2005.

[27] Y. Li, J. Ren, and J. Wu, "Quantitative measurement and design of source-location privacy schemes for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 7, pp. 1302–1311, 2012.

[28] A. A. Pirzada and C. McDonald, "Establishing trust in pure ad-hoc networks," in *Proceedings of the the 27th Australasian Conference on Computer Science (ACSC '04)*, pp. 47–54, 2004.

[29] M. Johnson, M. Healy, P. Van De Ven et al., "A comparative review of wireless sensor network mote technologies," in *Proceedings of the IEEE Sensors 2009 Conference*, pp. 1439–1442, October 2009.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 836056, 9 pages
http://dx.doi.org/10.1155/2013/836056

Research Article

A Game Theoretic Model for Wireless Sensor Networks with Hidden-Action Attacks

Yuxin Mao, Ping Zhu, and Guiyi Wei

School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China

Correspondence should be addressed to Yuxin Mao; maoyuxin@zjgsu.edu.cn

Received 3 May 2013; Accepted 9 July 2013

Academic Editor: Jaime Lloret

The unattended nature of wireless sensor networks makes them very vulnerable to an adversary's malicious attack. In this paper, we propose to apply game theory into solving the network security problem of wireless network. We explore game theory algorithms to model situation for wireless network with malicious nodes and investigate the attack and detection problem by modeling it as pairwise simultaneous game and spatial structured game. We consider the relationship between the nodes in a wireless sensor network to formulate the game and give the game theory algorithms in detail. We also evaluate the approach with a simulation experiment and analyze the simulation results in detail. We argue that the approach is able to support secure end-to-end communication in wireless sensor networks.

1. Introduction

The unattended nature of wireless sensor networks (WSNs) makes them very vulnerable to an adversary's malicious attacks. An adversary can physically compromise a subset of nodes in a WSN to eavesdrop information. The compromised nodes (or malicious nodes) become *black holes* [1] in the network. Those black holes in a WSN raise hidden-action attacks [2] to reduce the performance of the network or even destroy the network. Therefore, network security is an important issue for WSNs. In a WSN with malicious nodes, there are obvious conflicts between malicious nodes and normal nodes. As a branch of applied mathematics, game theory is concerned with how rational entities make decisions in a situation of conflict, which has been used primarily in economics. Game theory aims to model situations in which decision makers have to make specific actions that have mutual—possibly conflicting—consequences [3]. In the context of WSNs, game theory may be used to form cooperation schemes among entities in a competitive environment, for example, power control, routing, and resource allocation. Game theory can be applied to model the situation where there are malicious nodes in a WSN.

In this paper, we try to explore game theory to model situation for wireless network with malicious nodes and solve the problem of secure wireless communications by using a game-based approach. The main contributions of this paper are summarized as follows: (1) a formal representation model for simulating a WSN with malicious nodes and hidden-action attacks by using game theory is given; (2) several efficient game-based algorithms are proposed to support reliable and secure wireless communications against the attacks of malicious nodes in the WSN. The remaining of the paper is organized as follows. In Section 2, we first illustrate the system model including the network model as well as the problem statement. In Section 3, we present the game theoretic formulation for a WSN with malicious nodes. Game theory is used to solve the problem of reliable and secure wireless communications against the active attacks of malicious nodes in WSN. Moreover, we present the game theory algorithms in Section 4. In Section 5, we also evaluate the approach with a simulation experiment and analyze the simulation results in detail. Section 6 gives an overview of the related works. Section 7 concludes the paper with an outlook on future research directions.

2. Network Model and Problem Statement

2.1. Network Model. In this paper, we consider a relatively simple WSN. Consider $I = \{1, 2, \ldots, k\}$, and assume a WSN

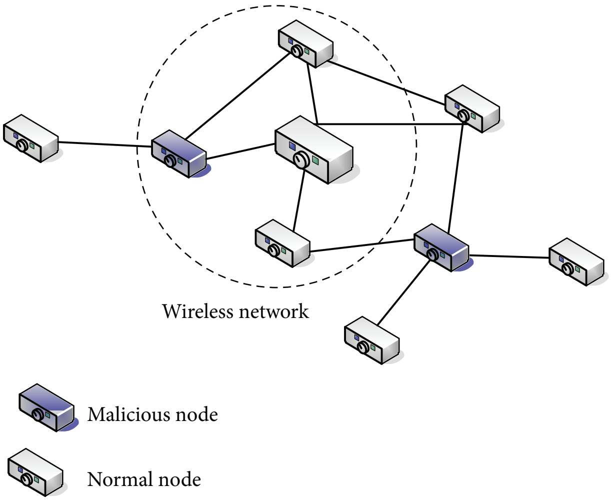


Figure 1: A wireless sensor network with malicious nodes.

with k nodes (see Figure 1). Each node has a unique identity i ($i \in I$) in the network. We assume the locations of the nodes are relatively static rather than dynamic. Each node in the network has at least one neighbor node. Each node in the network is battery powered and has limited computation and wireless communication capabilities. Assume the initial energy value of each node in the network is η. Each normal node in the network has four kinds of actions.

(1) *Forward* means a node sends a data packet to another node in the network.

(2) *Receive* means a node receives a data packet from another node in the network.

(3) *Detect* means a node receives a data packet from another node in the network and analyzes the pattern of the packet to find out any abnormality.

(4) *Sleep* means a node turns off its antenna and does not take any other actions.

Each action (except sleep) of a node will consume certain of energy. Therefore, either a normal node or a malicious node intends to sleep periodically, in order to save energy. An adversary is able to compromise a node or even physically capture a node. Therefore, there are a number of malicious nodes in the network. Assume the number of the malicious nodes is h ($0 < h \ll k$). Each malicious node in the network has four kinds of actions.

(1) *Forward*.

(2) *Receive*.

(3) *Jam* means a malicious node sends an abnormal data packet to another node in the network, in order to block the channel of the node.

(4) *Sleep*.

Normal nodes do not have knowledge of the instantaneous channel state information (CSI) of malicious nodes, but they know their distribution, and vice versa. Moreover, we assume that the number of the malicious nodes is much smaller than the number of the sensor nodes in the network; that is, $h \ll k$. It makes sense that a network is out of control if the number of the malicious nodes is very large. Another important assumption is that the type or identity (normal or malicious) of the nodes is unknown or blind. A node only knows the identity of itself.

2.2. Problem Statement. The major task of a normal node in the network is to transmit data to other nodes by routing. However, there is an additional task for normal nodes in a WSN with malicious nodes, that is, to detect the abnormal behaviors of malicious nodes. The major task of a malicious node is to block as many data packets of normal nodes as possible by attacking. A malicious node has two kinds of attack methods, passive methods like eavesdropping and active methods like jamming and DoS attack. In WSN, passive attacks of malicious nodes are hard to be detected, while active attacks have a risk of being detected. In this work, we mainly focus on the positive ones. Malicious nodes, in order to allay suspicions or save energy, only take attacks intermittently.

Moreover, we assume that detecting malicious nodes is an accumulative process to a normal node. Every detecting action of a normal node may gain a piece of evidence. When a normal node allocates enough evidence, it could then perform intrusion detection [4, 5] and locate malicious nodes.

3. Game Theoretic Formulation

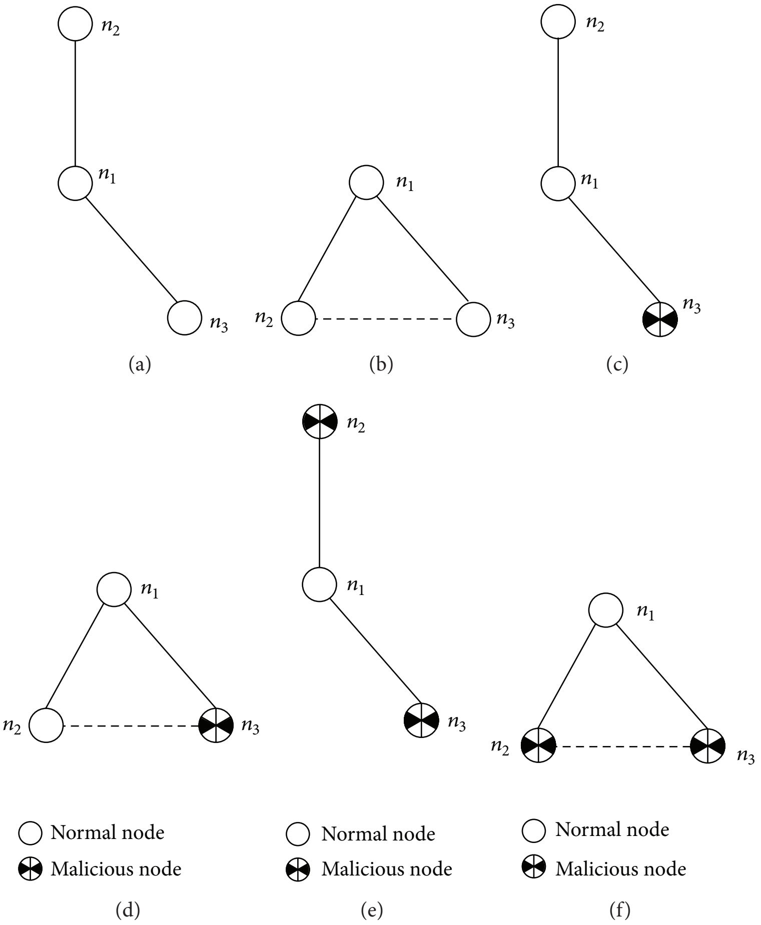
In this section, we investigate the attack and detection problem by modeling it as a simultaneous game. In this game, we consider the relationship between every pair of nodes in the network. We try to formulate different cases for this game in detail.

3.1. Pairwise Simultaneous Game. Let us consider a WSN with two nodes, a normal node n and a malicious node m. It is a pairwise simultaneous game between the two nodes, where both players move simultaneously. Each player has four strategies (see Table 1). Here γ is the direct income for a node to send out a data packet successfully; that is, the packet is received by another node. α_F is the probability that a normal node sends out a data packet successfully ($1 > \alpha_F > 0$). α_J is the probability that a malicious node sends out a jamming packet successfully ($1 > \alpha_J > \alpha_F$). It makes sense that jamming is more likely to be successful than forwarding. Here p is a punishment to a detected jamming action of the malicious node, and s is a stimulus to a successful jamming detection of the normal node ($p > 0, s > 0$).

We formulate the energy consumption in the game by some constants. β_1 is the cost to receive a packet (receiving or detecting, $\beta_1 > 0$), and $\Delta\beta_1$ is the additional cost for detecting attacks ($\Delta\beta_1 > 0$). β_2 is the cost to send out a packet (forwarding or jamming, $\beta_2 > 0$), and $\Delta\beta_2$ is the additional cost for jamming because jamming action consumes more energy compared with normal forwarding action ($\Delta\beta_2 > 0$).

Table 1: The result of two nodes' game (one normal node versus one malicious node).

	Normal node (n)			
	Forward	Detect	Receive	Sleep
Malicious node (m)				
Forward	$\{-\beta_2, -\beta_2\}$	$\{-\beta_2, -(\beta_1 + \Delta\beta_1)\}$	$\{\alpha_F\gamma - \beta_2, -\beta_1\}$	$\{-\beta_2, 0\}$
Jam	$\{-(\beta_2 + \Delta\beta_2), -\beta_2\}$	$\{-p - (\beta_2 + \Delta\beta_2), s - (\beta_1 + \Delta\beta_1)\}$	$\{\alpha_J\gamma - (\beta_2 + \Delta\beta_2), -\beta_1\}$	$\{-(\beta_2 + \Delta\beta_2), 0\}$
Receive	$\{-\beta_1, \alpha_F\gamma - \beta_2\}$	$\{-\beta_1, -\beta_1\}$	$\{-\beta_1, -\beta_1\}$	$\{-\beta_1, 0\}$
Sleep	$\{0, -\beta_2\}$	$\{0, -\beta_1\}$	$\{0, -\beta_1\}$	$\{0, 0\}$



(a) (b) (c)

(d) (e) (f)

Figure 2: Different patterns for local subgames.

As we have mentioned before, nodes do not know the identity of each other in our network model. Even a malicious node is not able to confirm one of its neighbor nodes as its ally. Therefore, there is also a game between two malicious nodes, m_1 and m_2. The result of two malicious nodes' game is shown in Table 2.

Similarly, there is a game between two normal nodes, n_1 and n_2. The result of two normal nodes' game is illustrated in Table 3.

3.2. Spatial Structured Game. Let us consider a more complex situation. There are a number of subgames between different pairs of nodes (either normal or malicious) in the whole network. We could reduce the complete game into a number of two nodes' games. For each two nodes' game, we could get a result by the formulation in Section 2.2 and then synthesize the result of the complete game by the result of all the subgames. Considering a part of a WSN with three nodes (n_1, n_2, n_3), we could get a series of patterns according to the network topology (see Figure 2).

The strategy set of a normal node is $S_{\text{normal}} = \{F, D, R, S\}$, and the strategy set of a malicious node is $S_{\text{malicious}} = \{F, J, R, S\}$. The meanings of F, D, J, R, and S are represented in Table 1. We use $*$ to denote any action in S_{normal} or $S_{\text{malicious}}$. Moreover, we use $-X$ to denote any action except X in S_{normal} or $S_{\text{malicious}}$. The payoff function for n_i in the subgame between n_i and n_j is denoted by u_{n_i,n_j}.

Table 2: The result of two nodes' game (one malicious node versus one malicious node).

	Malicious node (m_1)			
	Forward	Jam	Receive	Sleep
Malicious node (m_2)				
Forward	$\{-\beta_2, -\beta_2\}$	$\{-\beta_2, -(\beta_2 + \Delta\beta_2)\}$	$\{\alpha_F\gamma - \beta_2, -\beta_1\}$	$\{-\beta_2, 0\}$
Jam	$\{-(\beta_2 + \Delta\beta_2), -\beta_2\}$	$\{-(\beta_2 + \Delta\beta_2), -(\beta_2 + \Delta\beta_2)\}$	$\{\alpha_J\gamma - (\beta_2 + \Delta\beta_2), -\beta_1\}$	$\{-(\beta_2 + \Delta\beta_2), 0\}$
Receive	$\{-\beta_1, \alpha_F\gamma - \beta_2\}$	$\{-\beta_1, \alpha_J\gamma - (\beta_2 + \Delta\beta_2)\}$	$\{-\beta_1, -\beta_1\}$	$\{-\beta_1, 0\}$
Sleep	$\{0, -\beta_2\}$	$\{0, -(\beta_2 + \Delta\beta_2)\}$	$\{0, -\beta_1\}$	$\{0, 0\}$

Table 3: The result of two nodes' game (one normal node versus one normal node).

	Normal node (n_1)			
	Forward	Detect	Receive	Sleep
Normal node (n_2)				
Forward	$\{-\beta_2, -\beta_2\}$	$\{-\beta_2, -(\beta_1 + \Delta\beta_1)\}$	$\{\alpha_F\gamma - \beta_2, -\beta_1\}$	$\{-\beta_2, 0\}$
Detect	$\{-(\beta_1 + \Delta\beta_1), -\beta_2\}$	$\{-\beta_1, -\beta_1\}$	$\{-\beta_1, -\beta_1\}$	$\{-\beta_1, 0\}$
Receive	$\{-\beta_1, \alpha_F\gamma - \beta_2\}$	$\{-\beta_1, -\beta_1\}$	$\{-\beta_1, -\beta_1\}$	$\{-\beta_1, 0\}$
Sleep	$\{0, -\beta_2\}$	$\{0, -\beta_1\}$	$\{0, -\beta_1\}$	$\{0, 0\}$

Assume that n_1 is always a normal node in this part and there are four cases for the identities of n_2 and n_3.

In pattern (a) and pattern (b), both neighbors of n_1 are normal. There is a subgame between n_1 and n_2. There is another subgame between n_1 and n_3. Due to the broadcast character of WSN, we shall not calculate the cost of a node for more than one time when we compute the payoff of the node that plays subgames with several nodes at the same time.

In these patterns, we have the following exception rules to the game result in Table 3.

(1) If $u_{n_1,n_2}(F,R) = \alpha_F\gamma - \beta_2$, then $u_{n_1,n_3}(F,*) = 0$, and vice versa (Rule 1).

(2) If $u_{n_1,n_2}(F,*) = -\beta_2$, then $u_{n_1,n_3}(F,-R) = 0$ and $u_{n_1,n_3}(F,R) = \alpha_F\gamma$, and vice versa (Rule 2).

(3) If $u_{n_1,n_2}(D,*) = -(\beta_1 + \Delta\beta_1)$ or $u_{n_1,n_2}(D,*) = -\beta_1$, then $u_{n_1,n_3}(D,*) = 0$, and vice versa (Rule 3).

(4) If $u_{n_1,n_2}(R,*) = -\beta_1$, then $u_{n_1,n_3}(R,-F) = 0$, and vice versa (Rule 4).

In pattern (c) and pattern (d), n_1 has only one malicious neighbor (n_3), and the its other neighbor (n_2) is normal. Besides the constraint in patterns (a) and (b), there is one more constraint for the three nodes in these two patterns. In pattern (c), the data that flows from n_2 to n_1 is affected by n_3. Therefore, when we compute the subgame between n_1 and n_2, we should consider this constraint. In pattern (d), there is one more edge between n_2 and n_3. That is, n_3 is a common neighbor of n_1 and n_2. It means that n_3 has effects on both n_1 and n_2, or both the data that flows from n_1 to n_2 and the one from n_2 to n_1 are affected by n_3.

In pattern (c), we have the following exception rules to the game results in Tables 1 and 3.

(1) Rule 1.

(2) Rule 2.

(3) If $u_{n_1,n_2}(D,*) = -(\beta_1 + \Delta\beta_1)$ or $u_{n_1,n_2}(D,*) = -\beta_1$, then $u_{n_1,n_3}(D,*) = 0$, and vice versa (Rule 5).

(4) If $u_{n_1,n_2}(R,*) = -\beta_1$, then $u_{n_1,n_3}(R,-F) = 0$, and vice versa (Rule 6).

(5) If $u_{n_3,n_1}(J,R) = \alpha_J\gamma - (\beta_2 + \Delta\beta_2)$, then $u_{n_2,n_1}(F,R) = -\beta_2$ (Rule 7).

In pattern (d), we have one more exception rule.

(6) If $u_{n_3,n_1}(J,R) = \alpha_J\gamma - (\beta_2 + \Delta\beta_2)$, then $u_{n_1,n_2}(F,R) = -\beta_2$ (Rule 8).

In pattern (e) and pattern (f), both neighbors of n_1 are malicious. Similarly, with pattern (c) and pattern (d), a malicious node will affect the subgame of its neighbor. In pattern (e), the data that flows from n_2 to n_1 is affected by n_3, while the data that flows from n_3 to n_1 is also affected by n_2. In pattern (f), both the data flow from n_1 to n_2 and the one from n_2 to n_1 are affected by n_3, while both the data that flows from n_1 to n_3 and the one from n_3 to n_1 are affected by n_2.

In pattern (e), we have the following exception rules to the game results in Table 2.

(1) Rule 1.

(2) Rule 2.

(3) If $u_{n_1,n_2}(R,*) = -\beta_1$, then $u_{n_1,n_3}(R,R) = 0$ and $u_{n_1,n_3}(R,S) = 0$, and vice versa (Rule 9).

(4) If $u_{n_2,n_1}(J,R) = \alpha_J\gamma - (\beta_2 + \Delta\beta_2)$, then $u_{n_3,n_1}(F,R) = -\beta_2$, and vice versa (Rule 10).

In pattern (f), we have one more exception rule.

(5) If $u_{n_2,n_1}(J,R) = \alpha_J\gamma - (\beta_2 + \Delta\beta_2)$, then $u_{n_1,n_3}(F,R) = -\beta_2$, and vice versa (Rule 11).

If we take the whole network as an undirected graph, each edge denotes a subgame between two nodes. We could compute the game result of each edge in the graph by traversing the graph with a certain traversal algorithm (e.g., BFS or DFS).

```
Input: A wireless node $n_0$.
Output: A payoff value $p_0$.
get the neighbour nodes of $n_0$ as $NB = \{n_1, n_2, \dots, n_k\}$
for each node $n_i$ in $NB$ $(i = 0, 1, \dots, k)$
    perform a two nodes' game between $n_0$ and $n_i$
end loop
construct a many-to-many game for $n_0$ and $NB$
compute payoff $p_0$ for $n_0$ under the restriction of the rules
in Section 3.2
return$p_0$
```

Algorithm 1: The game algorithm for a wireless node.

4. Game Theory Algorithms

We could formulate the game for a WSN given in Section 3 by the following algorithms. First, let us consider the case for one specific wireless node (either normal node or malicious node). Algorithm 1 is proposed to compute the payoff of a specific wireless node.

The algorithm is trivial. The payoff of a specific node is impacted by the roles of its neighbors. In order to compute the payoff of a specific node, we have to consider the relationship between the node and each of its neighbors.

Then we could compute the payoffs of the whole network based on Algorithm 1. The algorithm for computing global payoffs is given as follows.

Algorithm 2 is also easy to be understood. The process of computing global payoffs is reduced to traversing an undirected graph by BFS. For each node, we just perform a single-node payoff computation, and then we get the information for the whole network. To a given WSN, we shall perform a series of computations to reach convergence.

5. Simulation and Evaluation

5.1. Simulation Setting. In this section, we construct simulation to evaluate the performance of the proposed approach. We consider a WSN with N players (wireless nodes). Player i is represented as a vertex v_i of a graph $G(V, E)$, with $v_i \in V$. An interaction between two players i and j is represented by an undirected edge $e_{ij} \in E$. The number of neighbors of player i is the degree k_i of vertex v_i. The average degree of the network is denoted by $\langle k \rangle$. The terms vertex, individual, participant, and player are used interchangeably in this section, likewise for edge, interaction, and link. Each malicious node can take one of the four strategies: $\{F, D, R, S\}$, while each normal node can take one of the four strategies: $\{F, J, R, S\}$.

The *BA scale-free* network [6] is adopted to represent the population structure of our WSN, which is constructed according to the "growth" and "preferential attachment" mechanisms. Starting from m_0 fully connected nodes, a new node with m $(m \leq m_0)$ edges is added to the system at every step. The new node links to m different nodes by a "preferential attachment" mechanism. The probability of connecting to an existing node i is proportional to its degree; that is, $p_i = k_i / \sum_j k_j$, where j runs over all existing nodes and k_i is the degree of node i. After t time steps, this algorithm produces a graph with $N = t + m_0$ vertices and mt edges.

Let $S(t) = (s_1(t), s_2(t), \dots, s_N(t))$ denote a configuration of the population strategies $s_i(t) \in \{F, D, J, R, S\}$ at time step t; the global synchronous system dynamics leads to $S(t+1)$ by simultaneously updating all the players' strategies according to the chosen rule, such as Fermi update rule [7]. Here, by synchronous, we mean that player's strategy will not change (even if it had already changed during its strategy updating process at time step t) until all the other $N-1$ players complete their strategy updating processes.

The game process is the same as the standard evolutionary game. At each step, all nodes are synchronously updated according to a strategy update rule. Note that in realistic WSNs, there may be environmental noise, which influences individuals' decisions (e.g., strategy mutation). Therefore, we adopt the Fermi updating rule, which considers environmental noise. When player i updates its strategy, it will first select a neighbor j out from all its k_i neighbors at random and then adopt j's strategy with the probability $H_{i \to j} = 1/(1 + e^{(P_i - P_j)/\kappa})$.

Here, κ characterizes the environmental noise, including bounded rationality, individual trials, and errors in decision; $\kappa \to \infty$ leads to neutral (random) drift whereas $\kappa \to 0$ corresponds to the imitation dynamics, where player j's strategy replaces player i's whenever $P_j > P_i$. For finite value of κ, the smaller the κ is, the more likely the fitter strategy is to replace the less fit one; thus the value of κ indicates the intensity of selection.

Finally, Tables 1–3 describe the payoffs of normal nodes and malicious nodes; Figure 2 depicts all the six exception rules in the network.

5.2. Experiment Results. In the following simulations, we study the effect of β_1 (cost of receiving a package), β_2 (cost of sending out a package), $\Delta\beta_1$ (additional cost of detecting attacks), $\Delta\beta_2$ (additional cost of jamming), p (punishment to a detected jamming action of the malicious node), and s (stimulus to a successful jamming detection of the normal node). α_F is set to 75%, and α_J is set to 100% (i.e., whenever a malicious node decides to jam the network, it will succeed). Besides, γ the direct income for a node to send out a data packet successfully is set to unit payoff. All the simulations were carried out on a *BA scale-free* network with network size $N = 1000$ and $m = m_0 = 4$; therefore, the average degree $\langle k \rangle = 8$. Initially, normal nodes and malicious nodes are randomly distributed among the population with probability of 90% and 10%, respectively. In addition, for normal nodes, the four strategies, that is, F, D, R, and S, are assigned the same probability, that is, 25%; for malicious nodes, the four strategies, that is, F, J, R, and S, are also assigned the same probability. The equilibrium frequencies of malicious jamming nodes are obtained by averaging over 3000 generations after a transient time of 10,000 generations. Each piece of data is averaged over 1000 runs on 1000 different networks.

```
Input: A WSN with a collection of nodes $N = \{n_0, n_1, \ldots, n_k\}$, a source node $n_0$, and a
collection of malicious nodes $M = \{m_i, m_{i+1}, \ldots, m_h\}$, where $n_0 \in N, M \subset N$.
Output: A collection of payoff values $P = \{p_0, p_1, \ldots, p_k\}$ for $N$, where $p_i$ is the payoff of
$n_i$ $(i = 0, 1, \ldots, k)$.
Assign an initial action to each $n_i$ in $N$ and the action of $n_0$ is Forward
Set $n_0$ as the root node
Perform BFS by starting with $n_0$
for each intermediate node $n_i$ in BFS
    compute payoff for $n_i$ by using Algorithm 1
    get $p_i$ for $n_i$
end loop
Return $P = \{p_0, p_1, \ldots, p_k\}$
```

Algorithm 2: The game algorithm for a WSN with malicious nodes.

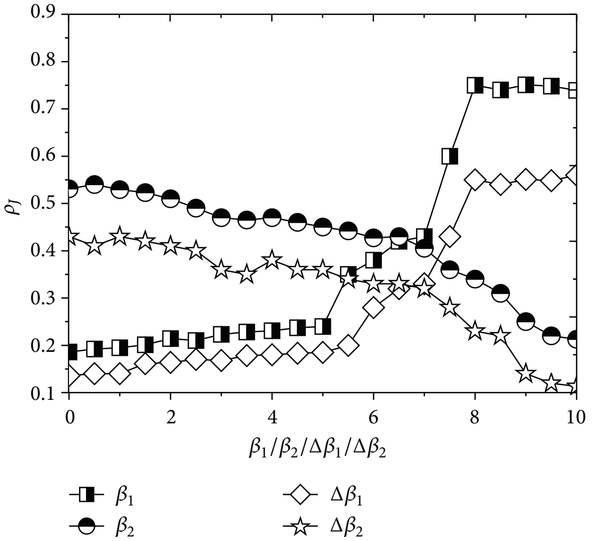


Figure 3: Effect of package-receiving cost, package forwarding cost, jamming detecting cost, and jamming cost on the proportion of malicious jamming nodes.

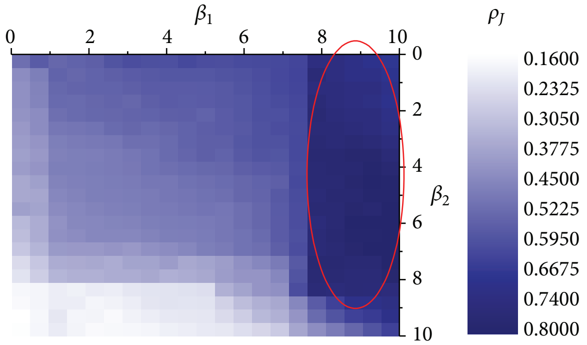


Figure 4: Coaction effect of forwarding and receiving on the proportion of malicious jamming nodes.

5.2.1. Action Cost Effect. Figure 3 shows the relationship between network action (F, D, J, R, and S) cost and ρ_J, where $\rho_J = N_m^{J^*}/N_m$ is the ratio of the number of malicious nodes that take the jamming action N_m^{J*} to the total number of malicious nodes N_m in the equilibrium state. As previously referred, the initial proportion of malicious nodes $N_m/N = 10\%$ and of N_m^F (number of initial malicious forwarding nodes), N_m^R (number of initial malicious receiving nodes), N_m^J (number of initial malicious jamming nodes), and N_m^S (number of initial malicious sleeping nodes) is assigned 25%, respectively. We can see that by increasing package-receiving cost or additional detecting cost, ρ_J increases; however, when increasing package-sending cost or additional jamming cost, ρ_J decreases. We argue that this result is expected. It makes sense that normal nodes incline to perform package sending rather than detecting when the detecting cost increases. As a result, malicious nodes are more likely to perform jamming because there are not enough normal nodes detecting their attacks. Moreover, malicious nodes incline to be (or pretend to be) "normal" or when the cost of attacks increases.

5.2.2. Coaction Effect. To further explore the effect of network action cost, we jointly study the coaction of forwarding and receiving. The result is shown in Figure 4. The value of ρ_J is represented in the form of color depth. As Figure 3 indicated, malicious nodes that take the jamming action will emerge when β_1 is large and β_2 is small. On the other hand, when β_1 is small and β_2 is large, the self-evolved network can inhibit hidden-action attack, for example, jamming, effectively. Figure 4 just confirms our prediction. As we can see, the deepest blue area locates in the northeast corner of the axes and the lightest blue area locates in the southwest corner of the axes. We can also find that β_1 performs a bigger weight than β_2 on the equilibrium proportion of malicious nodes that take the jamming action. We also find that when β_1 is large enough, variation of β_2 has little effect on ρ_J (see the red-circled area in Figure 4). This result is in accordance with Figure 3, where the $\beta_1 - \rho_J$ curve is steeper than the $\beta_2 - \rho_J$ curve, indicating that change of β_1 is much more influential.

5.2.3. Stimulation and Punishment Effect. Next, we explore the impacts of detecting stimulation s and jamming punishment p on ρ_J. The result is shown in Figure 5. It is indicated

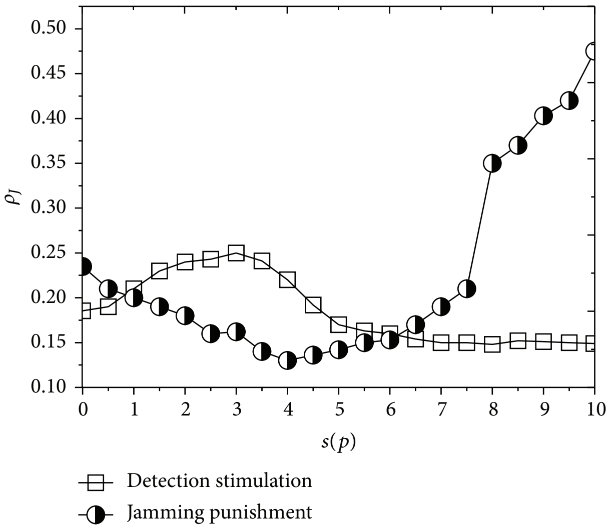


FIGURE 5: Effect of detecting stimulation and jamming punishment on the proportion of malicious jamming nodes.

that, for detecting stimulation, as s increases, ρ_J firstly increases to a local maximum and then decreases to a stable value. For jamming punishment, as p increases, ρ_J firstly decreases to a local minimum and then increases to about 50%. We argue that both of two curves are just as we expected. For detecting stimulation, when s is small, it is not enough to encourage normal nodes to detect malicious jamming nodes until it reaches a threshold value, after which more normal nodes joined in for detection, and thus ρ_J decreases. However, when s is large, increase of detecting stimulation makes no sense. Because the number of malicious jamming nodes is small, the total detecting payoff is limited, which is unable to attract more normal nodes to join and perform detecting. For jamming punishment, when p is small, the malicious nodes that take the jamming action are efficiently constrained. However, when p is large, the punishment cost exceeds the income for punisher; thus the number of normal nodes that take the detecting action decreases and results in the increase of ρ_J.

Similarly, we also study the joint effect of s and p on the proportion of malicious nodes that take jamming action. The result is given in Figure 6. According to the result of Figure 5, a large ρ_J will appear when p is large and s is around 3. However, as Figure 6 depicts, the maximum of ρ_J has an apparent drift towards larger value of s (see the blue-circled area). This can be explained as follows: when the jamming punishment p is large, normal nodes are unwilling to perform detecting, because the punishment cost exceeds the income for punisher. Therefore, these normal nodes will wait until the detection stimulation increases to an acceptable threshold. The orbit of drift is shown as an arrowed curve in Figure 6. The black-circled area indicated the smallest ρ_J, which is also in accordance with Figure 5. Like that in Figure 4, here we argue that the jamming punishment p performs a bigger weight than detecting stimulation s on the equilibrium proportion of malicious nodes that take the jamming action.

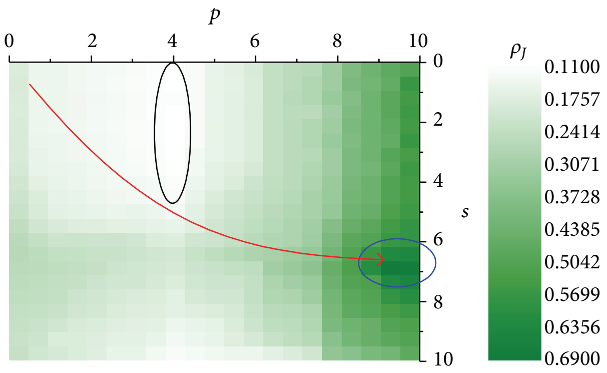


FIGURE 6: Joint effect of detecting stimulation and jamming punishment on the proportion of malicious nodes taking the jamming action.

6. Related Works

Applying game theory into improvement the QoS of WSN or wireless network is a new research direction, especially the games for the security in WSN or wireless network. There have been a few research efforts in this field. However, there are still many issues worthy of further exploration.

Kodialam and Lakshman in [8] model a zero-sum game between the intruder and service provider of the network. The objective of the intruder is to inject a malicious packet in the network at some node with node t as the target. The intrusion is successful when the packet reaches the target and unsuccessful when it does not. To protect the nodes from the attack, the service provider is allowed to sample the packets flowing through the links on the network. The optimal solution for this game is the min.-max. optimal solution, which is the Nash equilibrium for a zero-sum game. Agah et al. in [9] study a game formulation at the routing layer of a wireless network between malicious nodes that do not forward incoming packets and an intrusion detector residing at the base station. They model this scenario as a repeated game, where the IDS uses the history of nodes' collaboration to determine paths comprising malicious nodes. The proposed protocol for the repeated game shows a correlation between network size and successful intrusion detection, where detection success rate increases with higher percentage of malicious nodes. Kamhoua et al. in [10] investigate a situation that it is cost effective to freely participate in the security mechanism or protect its privacy depending for each node on the fact that if that node believes or trusts that all other nodes or at least a minimum number of other nodes will do the same. They model a trustable dilemma for autonomous multihop networks by using the mathematical framework of game theory and evolutionary game theory. The well-known stag hunt game is used as their basic game model. They present the interconnection between cooperation, trust, privacy, and security in a network. However, they only consider the game for normal nodes in the network rather than the game between normal nodes and malicious nodes.

Khouzani et al. in [11] develop a zero-sum dynamic game model and investigate the structural properties of the saddle-point strategies. They consider a game with two players (normal nodes and malicious nodes) and three different states with normal nodes. Their research efforts show that saddle-point strategies are simple threshold-based policies, and hence a robust dynamic defense is practicable. They assume there is no competition among normal nodes or malicious nodes in the network, which is not feasible to many situations in WSNs. Sagduyu et al. in [12] present a class of jamming games played at the MAC layer of wireless network among a set of transmitters and jammers. They address the incomplete information or uncertainty in games compared with existing works. The equilibrium strategies resulting from the jamming games characterize the expected performance under DoS attacks and motivate robust network protocol design to support secure wireless communications. Although they consider the conflicts among malicious nodes to be incomplete information, they have not taken the conflicts of normal nodes into consideration. Jaramillo and Srikant in [13] discuss the problem that packet collisions and interference may make cooperative nodes appear selfish sometimes, generating unnecessary and unwanted punishments in wireless ad hoc networks. They present a robust mechanism to imperfect measurements, which is collusion resistant and can achieve cooperation among nodes. However, their method does not satisfy the situation where there are malicious nodes in the network. Yan et al. in [14] proposed a penalizing mechanism to prevent the noncooperative selfish behavior of decreasing the contention window without permission based on repeated game theory for WSNs. A Contention Window Select Game is defined, and in this game each sensor node selects its own contention window to control the access probability.

Moreover, there are some research efforts about using other evolutionary algorithms rather than game theory to sovle the security problem in WSNs or wireless networks. For example, Alrajeh et al. in [15] present an adaptive secure routing protocol based on bioinspired technique termed as ant colonization for WSNs. Their approach is able to select optimal paths from source to destination by ensuring adaptability, robustness, and security. However, their approach cannot suppress hidden-action attacks in WSNs. Hortos in [16] presents a cross-layer approach to WSN protocol design by applying a bioinspired evolutionary computational method to the functions of each protocol layer to improve the intrusion detection identification (IDID) performance. Genetic algorithms and ant colony optimization are used to solve the problem in the proposed approach.

Although many existing approaches have taken game theory to solve the security problem in WSNs or wireless networks, compared with existing works in this field, our approach considers the relationship between nodes in a WSN and gives a formal theoretic model for the game. We discuss different patterns in this game and represent them as rules to generate a more complete game model. We argue that such a representation model is relatively lacking in existing research efforts. Moreover, the algorithms proposed in this paper are not very complex and easy to be implemented in resource-constrained WSNs, compared to the existing approaches. In general, our approach is able to support secure wireless communication in WSNs with malicious nodes.

7. Conclusion

In this paper, we mainly present an approach to apply game theory into solving the network security problem of WSN. We try to explore game theory algorithms to model the situation for WSN with malicious nodes, in order to support reliable and secure wireless communications against the attacks of malicious nodes in the network. We consider the relationship between the nodes in a WSN to formulate the game and illustrate the game theory algorithms in detail. In order to verify the proposed approach, we also evaluate it with a simulation experiment and analyze the simulation results in detail. We argue that the approach is able to support secure end-to-end communication in WSNs with malicious nodes.

Future works may include (1) improving the efficiency of the algorithms to support different kinds of attacks rather than jamming, (2) evaluating the cost of the game theory algorithm for WSN, and (3) considering a more complex WSN model to evaluate the approach.

Acknowledgments

This work is partially supported by Grants from the NSFC Program (no. NSFC61003309), the Science and Technology Department of Zhejiang Province Program (no. 2010C13005, 2011C14024 and 2011C13006-1), the Key Science and Technology Innovation Team of Zhejiang Province Program (no. 2010R50041-03), the ZJNSF Program (no. LR12F02002). This work is also partially supported by the National Development and Reform Commission, China under Special Grants "The Operation System of Multimedia Cloud Based on the Integration of Telecommunications Networks, Cable TV Networks and the Internet".

References

[1] 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.

[2] T. Moore, "Countering hidden-action attacks on networked systems," in *Proceedings of the 4th Workshop on the Economics of Information Security*, Cambridge, Mass, USA, June 2005.

[3] D. Fudenberg and J. Tirole, *Game Theory*, MIT Press, 1991.

[4] Y. Zhang and W. Lee, "Intrusion detection in wireless adhoc networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 275–283, August 2000.

[5] K. Ioannis, T. Dimitriou, and F. C. Freiling, "Towards intrusion detection in wireless sensor networks," in *Proceedings of the 13th European Wireless Conference*, 2007.

[6] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[7] G. Szabó and C. Töke, "Evolutionary prisoners dilemma game on a square lattice," *Physical Review E*, vol. 58, no. 1, pp. 69–73, 1998.

[8] M. Kodialam and T. V. Lakshman, "Detecting network intrusions via sampling: a game theoretic approach," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 1880–1889, April 2003.

[9] A. Agah, K. Basu, and S. K. Das, "Prevention of DoS attack in sensor networks using repeated game theory," in *Proceedings of the International Conference on Wireless Networks (ICWN '05)*, pp. 29–36, 2006.

[10] C. A. Kamhoua, N. Pissinou, and K. Makki, "Game theoretic modeling and evolution of trust in autonomous multi-hop networks: application to network security and privacy," in *Proceedings of the IEEE International Conference on Communications (ICC '11)*, pp. 1–6, June 2011.

[11] M. H. R. Khouzani, S. Sarkar, and E. Altman, "A dynamic game solution to malware attack," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '11)*, pp. 2138–2146, April 2011.

[12] Y. E. Sagduyu, R. A. Berry, and A. Ephremides, "Jamming games in wireless networks with incomplete information," *IEEE Communications Magazine*, vol. 49, no. 8, pp. 112–118, 2011.

[13] J. J. Jaramillo and R. Srikant, "A game theory based reputation mechanism to incentivize cooperation in wireless ad hoc networks," *Ad Hoc Networks*, vol. 8, no. 4, pp. 416–429, 2010.

[14] 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, Shanghai, China, January 2011.

[15] N. A. Alrajeh, M. S. Alabed, and M. S. Elwahiby, "Secure ant-based routing protocol for wireless sensor network.," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 326295, 9 pages, 2013.

[16] W. S. Hortos, "Bio-inspired, cross-layer protocol design for intrusion detection and identification in wireless sensor networks," in *Proceedings of the 37th Conference on Local Computer Networks Workshops (LCN '12)*, pp. 1030–1037, 2012.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 904953, 10 pages
http://dx.doi.org/10.1155/2013/904953

Research Article

Lightweight and Scalable Intrusion Trace Classification Using Interelement Dependency Models Suitable for Wireless Sensor Network Environment

Dae-Ki Kang

Division of Computer & Information Engineering, Dongseo University, 47 Jurye-ro, Sasang-gu, Busan 617-716, Republic of Korea

Correspondence should be addressed to Dae-Ki Kang; dkkang@dongseo.ac.kr

Received 3 May 2013; Accepted 9 June 2013

Academic Editor: S. Khan

We present a lightweight and scalable method for classifying network and program traces to detect system intrusion attempts. By employing interelement dependency models to overcome the independence violation problem inherent in the Naive Bayes learners, our method yields intrusion detectors with better accuracy. For efficient and lightweight counting of n-gram features without losing accuracy, we use a k-truncated generalized suffix tree (k-TGST) for storing n-gram features. The k-TGST storage mechanism enables us to scale up the classifiers, which cannot be easily achieved by Support-Vector-Machine- (SVM-) based methods that require implausible computing power and resources for accuracy. Experimental results on a set of practical benchmark datasets show that our method is scalable up to 20-gram with consistent accuracy comparable to SVMs.

1. Introduction

Data mining algorithms have been widely used for classifying program traces (i.e., sequences of system calls or sequences of network packets) in intrusion detection tasks. For example, in a host-based intrusion detection task, a program trace is defined as a sequence of system calls that the program invokes during its execution, whereas in a network-based intrusion detection task, a program trace can be defined as a sequence of network packets that the program transmits during its execution. As a preprocessing step for data mining algorithms on intrusion detection tasks, the n-gram (n consecutive system calls in a trace) approach [1] has been widely used for featurization of system call sequences [2–5].

However, those n-gram approaches suffer from three critical problems when applied to intrusion detection tasks.

(1) *Dimensionality Issues.* Since the number of distinct system calls is usually as many as about 200, the number of distinct n-gram features increases drastically as n increases. For example, the number of distinct SunOS system calls is 183, and if a 20-gram approach is used, then the number of 20-gram features will be 183^{20}, which is impractical for real world applications. The data mining algorithms that have nonlinear space complexity (such as Support Vector Machines (SVMs) [6, 7]) severely suffer from this problem.

(2) *Overlap of Features.* When n-gram features are generated from an original trace using a fixed-length sliding window, one system call in the trace can be considered as many as n times in the worse case, in the resulting n-gram features [8, 9].

(3) *Violation of Independence Assumption.* If the resulting intrusion detectors rely on the statistical independence assumption among features (e.g., Naive Bayes), the above-mentioned way of generating overlapped features systematically breaks the assumption.

Against these backgrounds, we applied interelement dependency models [8, 10] of n-gram features to intrusion detection tasks and compared their performance with those of widely used data mining algorithms such as Naive Bayes with n-gram features and those of SVM with n-gram features. To overcome the curse of dimensionality problem, we adapted the k-truncated generalized suffix tree storage mechanism [11, 12] to index system call traces and to generate counts for each n-gram in an efficient way. Since the features with more order information (i.e., longer n-gram features) from an appropriate amount of input data sets usually contribute more to classification, it is important for an intrusion detection algorithm to be scalable with the length of n-grams.

Experimental results on host-based and network-based intrusion detection benchmark data sets show that the proposed method outperforms the Naive Bayes learner with n-gram features as input, which breaks the independence assumption, on intrusion detection tasks and shows comparable accuracies and false positive rates to those of SVMs with n-gram features. With the suffix tree storage mechanism, we tested the performance of the classifiers up to 20-gram in the experiments, which indicates the advantage of scalability of the proposed combination of the interdependency model of n-gram features and suffix tree storage mechanism over the other methods. We were able to perform the experiments with n-grams much longer than 20-grams, but because of lack of data sets and overfitting, the results were not so significant.

The rest of the paper is organized as follows: Section 2 describes our method, Section 3 presents the experimental results, and Section 4 summarizes and concludes this paper.

2. Method

First of all, we introduce Naive Bayes with n-gram features (NB n-gram), interdependency models of n-Grams (IM(n)), and SVM with n-gram features (SVM n-gram). After that, we explain suffix tree mechanism which is used to store n-gram features.

Before we describe each method, we formally define the intrusion detection problem as follows: Let $\Sigma = \{s_1, s_2, s_3, \ldots, s_m\}$ be a set of system calls, where $m = |\Sigma|$ is the number of system calls. Data set D can be defined as a set of labeled sequences (i.e., program traces) $\{\langle Z_i, c_i\rangle \mid Z_i \in \Sigma^*, c_i \in \{0,1\}\}$, where $Z_i = z_1, z_2, z_3, \ldots, z_l$ is an input sequence and c_i is a corresponding class label denoting 0 for "normal" label and 1 for "intrusion" label. Given the data set D, the goal of a learning algorithm is to find an intrusion detector $h : \Sigma^* \rightarrow \{0,1\}$ that maximizes given criteria such as accuracy, F-1 measure, detection rate, and false positive rate.

If a probabilistic model is applied for the intrusion detector h (e.g., Naive Bayes), then the probabilistic model P_h specifies for a sequence Z the probability $P_h(Z = z_1, z_2, z_3, \ldots, z_l)$ as follows.

(1) For each class c_i, estimate the probabilities $P_h(c_i)$ using all the sequences Z coupled with c_i.

(2) For a new sequence Z, assign the class c such that

$$c_h = \underset{c\in\{0,1\}}{\operatorname{argmax}}\, P_h\left(Z = z_1, z_2, z_3, \ldots, z_l \mid c\right) \cdot P_h(c). \quad (1)$$

2.1. Naive Bayes Classifier. One of the important assumptions in the Naive Bayes classifier as a host-based intrusion detector is that each system call of the sequence is independent of the other system calls given the class label. Therefore, as for Naive Bayes, the classification (shown in (1)) of a new sequence will be formulated as follows:

$$c_{\mathrm{NB}} = \underset{c\in\{0,1\}}{\operatorname{argmax}}\, P_h(c) \cdot \prod_i P_h\left(z_i \mid c\right). \quad (2)$$

When the Naive Bayes classifier is applied to text or protein sequence classification, it treats each document or protein sequence as a bag or set of words or letters that denotes amino acid [9, 13]. There are a few researches [15–17] that explore intrusion detection tasks with a bag or set of system calls, but most intrusion detection researches focus on the n-gram approach [4, 14, 18, 19].

2.2. Naive Bayes with n-Gram Features (NB n-Gram). Since it is difficult to deal with variable length sequences directly, each sequence $Z \in \Sigma^*$ is mapped into a finite n-dimensional feature vector (i.e., n-gram features). In host-based intrusion detection tasks, if we want to monitor a program's behavior, we consider the program's trace as a sequence. Thus, to generate n-gram features from an input trace, a sliding window of size n is applied to the trace, moving from the beginning of the trace to the end of the trace by one system call at each step, to generate a bag of n-gram features from the trace. Then, the probabilistic model for this n-gram representation is straightforward from (2):

$$c_{\mathrm{NB}\,n\text{-gram}} = \underset{c\in\{0,1\}}{\operatorname{argmax}}\, P_h(c) \cdot \prod_{i=1}^{l-n+1} P_h\left(z_i, \ldots, z_{i+n-1} \mid c\right), \quad (3)$$

where l is the length of the sequence.

There is one serious problem in the NB n-gram approach. When n-gram features are generated from an original trace using a sliding window, one system call in the trace can be considered and included as many as n times in the resulting n-gram features. This systematically violates the independence assumption of the Naive Bayes learning algorithm.

2.3. Interdependency Models of n-Grams (IM(n)). To overcome the previously mentioned problem, we applied interdependency models of n-grams [8, 10] for scalable n-gram-based intrusion detection. The applied method tried to explicitly model the dependencies among the elements inside an n-gram feature generated from a sequence.

Figure 1 shows the model that describes dependencies among six consecutive elements in a sequence.

Following the Junction Tree Theorem [20], the probabilistic model for IM(n) is as follows:

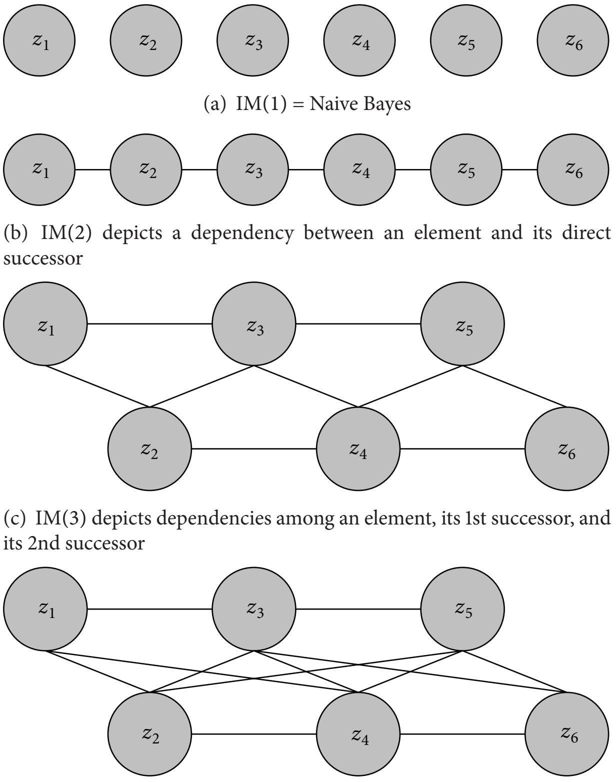
$$c_{\mathrm{IM}(n)} = \underset{c\in\{0,1\}}{\operatorname{argmax}}\, \frac{\prod_1^{l-n+1} P_h\left(z_i, \ldots, z_{i+n-1} \mid c\right)}{\prod_2^{l-n+1} P_h\left(z_i, \ldots, z_{i+n-2} \mid c\right)} P_h(c). \quad (4)$$

From Figure 1 and (4), it can be seen that the probabilistic graphical model of IM(n) is a Markov Network where the probabilistic distribution is obtained by dividing the product of the marginals of the maximal cliques (maximally connected subgraphs) in the graph by the product of the marginals of the separators (overlaps among cliques).

Algorithm 1 shows the pseudocode of the intrusion detection algorithm using the interdependency model.

```
IntrusionDetector($S$):
begin
  (1) Input: sequence data set $S = s_1, \ldots, s_n$ and interdependency model $\alpha$ as a
      probabilistic model
  (2) Learning: For each class $c_j$, estimate probabilities $P_\alpha(S = s_1, \ldots, s_n)$ of $\alpha(c_j)$
      based on $D$ that comprises the intrusion detector $h$
  (3) Testing: For a novel sequence $\widehat{S} = s_1, \ldots, s_n$, predict the classification $c(\widehat{S})$ as
      follows:
          $c(\widehat{S}) = \underset{c_j \in C}{\operatorname{argmax}} \left\{ P_\alpha(\widehat{S} = s_1, \ldots, s_n \mid c_j) P(c_j) \right\}$
end.
```

Algorithm 1: Intrusion detection algorithm with interdependency model.



(a) IM(1) = Naive Bayes

(b) IM(2) depicts a dependency between an element and its direct successor

(c) IM(3) depicts dependencies among an element, its 1st successor, and its 2nd successor

(d) IM(4) depicts dependencies among an element and its three closest successors

Figure 1: Graphical models that incorporate the dependencies among the six consecutive elements in a sequence.

2.4. Support Vector Machines with n-Gram Features (SVM n-Gram). For the comparison of IM(n)'s performance with the other data mining algorithm, we consider Support Vector Machines with n-grams as input.

It is of interest to compare IM(n) and NB n-gram with SVM n-gram, because, in contrast to Naive Bayes with n-gram features (NB n-gram), SVMs do not rely on the independence assumption between features.

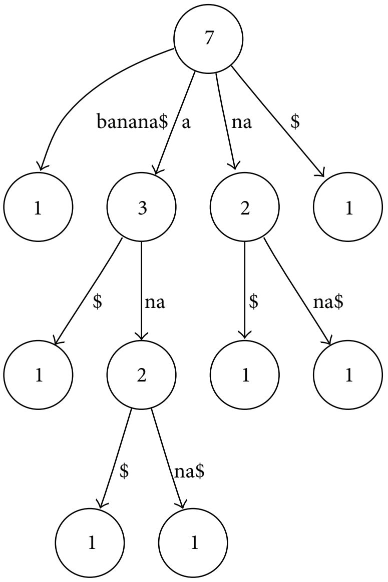


Figure 2: A suffix tree of a string "banana$". The number in each node represents the number of pattern occurrences.

However, the SVM algorithm suffers from the explosion of the number of features as n increases, because it takes at least $O(n^2)$ time and space to prepare a kernel matrix for SVM. In other words, since the SVM learning algorithm has nonlinear asymptotic complexity and due to the nature of n-gram attributes, as n increases, the number of attributes and the size of the working memory increase drastically (i.e., curse of dimensionality). In the actual experiment, because of these computational and memory requirements, we were only able to conduct the experiment of SVM n-gram for $n = 1, 2$, and 3.

2.5. k-Truncated Suffix Tree. A suffix tree is a data structure to index a string [11]. Figure 2 shows an example suffix tree for a string "banana$", where "$" denotes the end of the string. The number in each node represents the number of pattern occurrences. For example, in the string "banana$", "a" occurs three times and "na" occurs twice.

When the length of a string is l, then it takes $O(l)$ time [21] to build a suffix tree for the string. Once a suffix tree is generated, then it takes $O(m)$ time to find a pattern string with length m. Also, with edge-label compression, it only needs $O(l)$ space for a suffix tree.

In practice, to store multiple strings, a generalized suffix tree is used. A generalized suffix tree is a storage that contains

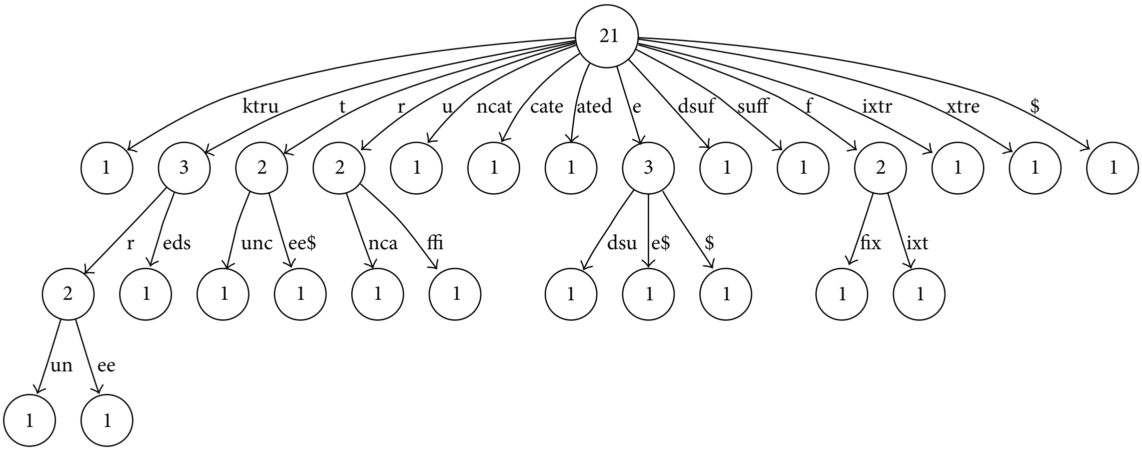


Figure 3: A 4-truncated suffix tree of "ktruncatedsuffixtree$".

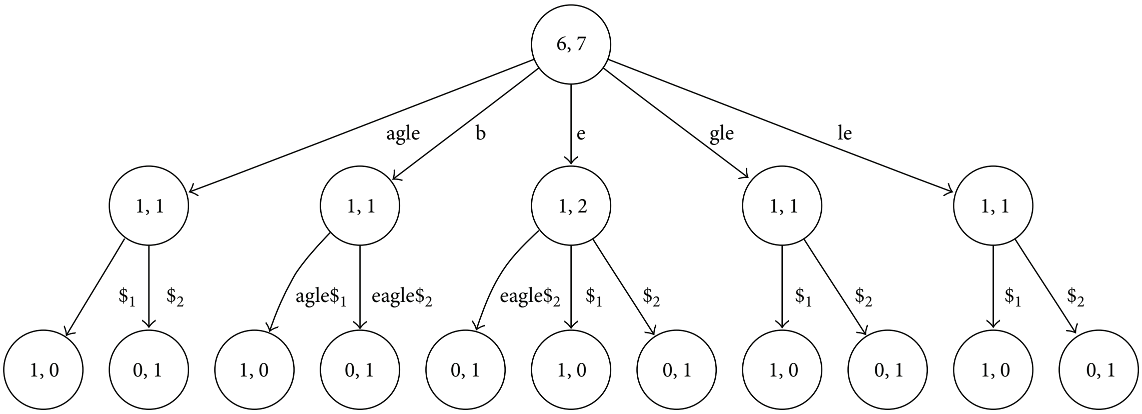


Figure 4: A generalized suffix tree of "bagle$\$_1$" and "beagle$\$_2$".

all suffixes of a set of strings [11]. An example of a generalized suffix tree for the strings "bagle$\$_1$" and "beagle$\$_2$" is shown in Figure 4.

Even with the suffix tree storage, it still takes a lot of memory to save the entire traces of system calls. However, in our application, it is not necessary to store a whole trace into a suffix tree. Instead, we store n-gram features into the generalized suffix tree, as shown in Figure 3, which are of interest for generating intrusion detectors.

3. Experimental Setup and Results

To evaluate the performance of interdependency models of n-grams (IM(n)), we compared its performance with Naive Bayes (NB), Naive Bayes with n-gram features (NB n-gram), and Support Vector Machines with n-gram features (SVM n-gram). For the experiment, we chose publicly available data sets from the University of New Mexico (UNM) [14] and MIT Lincoln Lab 1998 DARPA Intrusion Detection Evaluation Data Sets (MIT LL 1998) [22].

3.1. Data Sets

3.1.1. UNM Data Sets. The University of New Mexico (UNM) provides a number of system call data sets. Each data set corresponds to a specific attack or exploit. The data sets we tested are "live lpr", "live lpr MIT", and "denial of service" (DoS).

Table 1: The number of instances for each type of attack.

Attack	Positive	Negative	Total
Live lpr (`l_lpr`)	1001	1231	2232
Live lpr MIT (`l_lpr.MIT`)	1001	2703	3704
Denial of service (`stide`)	105	13726	13831

In UNM system call traces, each trace is an output of one program. Sometimes, one trace has multiple processes. In such cases, we have made as many sequences as the number of processes in the original trace. Thus, multiple sequences of system calls are made from one trace if the input trace has multiple processes in it. However, most traces have only one process and usually one sequence is created for each trace. Table 1 shows the number of original traces and the number of sequences for each data set.

There are two different mapping files in the UNM call traces we used for the experiment. One is Sun (live lpr and live lpr.MIT), and the other is Linux (denial of service). There are old and new Sun mapping files but only one system call is added to the new mapping file, so both can be easily

Table 2: Comparison of accuracy (A) and false positive rate (FP) of intrusion detectors generated by IM(n), NB n-gram, and SVM n-gram on the *UNM live lpr* data. The accuracies and false positive rates were estimated using 10-fold cross-validation. We calculated a 99% confidence interval on the accuracies and false positive rates. Note that when $n \geq 3$, SVM n-gram is infeasible because of computational and memory requirements.

N	IM(n)		NB n-gram		SVM n-gram	
	A	FP	A	FP	A	FP
1	84.09 ± 0.02	28.84 ± 0.02	84.09 ± 0.02	28.84 ± 0.02	**100.00 ± 0.00**	**0.00 ± 0.00**
2	99.78 ± 0.00	0.41 ± 0.00	98.30 ± 0.01	3.09 ± 0.01	99.96 ± 0.00	0.00 ± 0.00
3	99.96 ± 0.00	0.00 ± 0.00	99.01 ± 0.01	1.79 ± 0.01	N/A	N/A
4	99.96 ± 0.00	0.00 ± 0.00	99.60 ± 0.00	0.73 ± 0.00	N/A	N/A
5	99.96 ± 0.00	0.00 ± 0.00	99.82 ± 0.00	0.32 ± 0.00	N/A	N/A
6–8	**100.00 ± 0.00**	**0.00 ± 0.00**	**99.87 ± 0.00**	**0.24 ± 0.00**	N/A	N/A
9-10	99.96 ± 0.00	0.08 ± 0.00	99.82 ± 0.00	0.32 ± 0.00	N/A	N/A
11–20	99.96 ± 0.00	0.08 ± 0.00	99.78 ± 0.00	0.41 ± 0.00	N/A	N/A

converted. The Sun mapping file has a few duplicate system calls (e.g., "fstat", "stat", etc.), but we changed them such that each system call is unique.

3.1.2. MIT LL 1998 Data Sets.

In the MIT Lincoln Lab 1998 data sets [14], we used both seven weeks of training data and two weeks of testing data. The data comprises a detailed set of data files representing the state of a particular system over eight-hour daytime periods over the course of the nine weeks (seven for training and two for testing). Of interest to the experiments are the omnibus data files containing all system calls made during the collection period and the network traffic analysis files (distilled from raw network data) that identify inbound network connection attempts.

We explained the issues with cross-indexing the data files. The MIT Lincoln Lab data sets include omnibus files containing all system call traces. For each omnibus file, there is a separate, network traffic analysis data file that indicates inbound network connections to the system. Attack attempts are logged with the network data, which implies that labeling the training data requires cross-indexing this file with the system call trace file. The system call trace file identifies the source of each call using the process ID. Therefore, cross-indexing requires tracking the argument to the "exec" system call identifying the binary to be executed. Additionally, the timestamps from the network traffic analyzer do not exactly correspond to the execution timestamps from the operating system kernel. A tolerance of one second is arbitrarily chosen and permits the matching of a large majority of connection attempts with their corresponding server processes on the target system.

All processes detected that do not correspond to some network connection attempt identified in the trace are removed from consideration (since they cannot be classified), as are all calls attributed to a process ID for which an "exec" system call is not found.

As for the experimental setting, we both followed training and testing experiments as explained in the data set description and performed 10-fold cross-validation over the whole nine-week data sets.

3.2. Results.

We applied the following performance measures:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}},$$
$$\text{False Positive Rate} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \tag{5}$$

where TP is the number of true positives, FP is the number of false positives, TN is the number of true negatives, and FN is the number of false negatives.

Table 2 shows accuracies and false positive rates of the three algorithms (IM(n), NB n-gram, and SVM n-gram) on the "UNM live lpr" data. IM(n) shows the best performance when n is from 6 to 8. Its performance is comparable to that of Support Vector Machines (SVMs). Overall, IM(n) shows better performance than NB n-gram.

As for the "UNM live lpr" data, IM(n) and NB n-gram show the best performance when n is from 6 to 8. The best accuracy and false positive rate of IM(n) are 100.00 and 0.00, and the best accuracy and false positive rate of NB n-gram are 99.87 and 0.24. These results correspond to the widely known claim that "six is the magic number" [23]. However, for the other data sets ("UNM live lpr MIT" and "UNM denial of service") we have tested, the claim does not always hold.

Table 3 shows the results of the three algorithms for UNM live lpr MIT data. When n is from 17 to 20, IM(n) shows the best performance (accuracy is 99.97 and false positive rate is 0.00) over SVM (accuracy is 99.95 and false positive rate is 0.00) and NB n-gram (accuracy is 93.49 and false positive rate is 8.92).

In Table 4, which is for the UNM denial of service data, we can see that SVM with 2-grams shows the best performance (accuracy is 99.99 and false positive rate is 0.00), but the difference from the results of IM(n) is very marginal (accuracy is 99.93 and false positive rate is 0.00). Moreover, during the training stage, SVM learns its discriminative model by the Sequential Minimal Optimization algorithm [24], which takes more than several hours. In contrast, IM(n) takes only about one or two minutes and its performance is comparable to that of SVM.

Table 3: Comparison of accuracy (A) and false positive rate (FP) of intrusion detectors generated by IM(n), NB n-gram, and SVM n-gram on *UNM live lpr MIT* data. The accuracies and false positive rates were estimated using 10-fold cross-validation. We calculated a 99% confidence interval on the accuracies and false positive rates. Note that when $n \geq 3$, SVM n-gram is infeasible because of computational and memory requirements.

N	IM(n)		NB n-gram		SVM n-gram	
	A	FP	A	FP	A	FP
1	54.56 ± 0.02	62.26 ± 0.02	54.56 ± 0.02	62.26 ± 0.02	99.83 ± 0.00	0.14 ± 0.00
2	96.76 ± 0.01	4.44 ± 0.01	78.75 ± 0.02	29.12 ± 0.02	**99.95 ± 0.00**	**0.00 ± 0.00**
3	99.95 ± 0.00	0.00 ± 0.00	86.23 ± 0.01	18.87 ± 0.02	N/A	N/A
4	99.95 ± 0.00	0.00 ± 0.00	89.28 ± 0.01	14.69 ± 0.02	N/A	N/A
5	99.95 ± 0.00	0.00 ± 0.00	91.17 ± 0.01	12.10 ± 0.01	N/A	N/A
6	99.95 ± 0.00	0.00 ± 0.00	91.77 ± 0.01	11.28 ± 0.01	N/A	N/A
7	99.95 ± 0.00	0.00 ± 0.00	91.85 ± 0.01	11.17 ± 0.01	N/A	N/A
8	99.95 ± 0.00	0.00 ± 0.00	91.98 ± 0.01	10.99 ± 0.01	N/A	N/A
9	99.95 ± 0.00	0.00 ± 0.00	92.09 ± 0.01	10.84 ± 0.01	N/A	N/A
10	99.95 ± 0.00	0.00 ± 0.00	92.04 ± 0.01	10.91 ± 0.01	N/A	N/A
11	99.95 ± 0.00	0.00 ± 0.00	92.33 ± 0.01	10.51 ± 0.01	N/A	N/A
12	99.95 ± 0.00	0.00 ± 0.00	92.68 ± 0.01	10.03 ± 0.01	N/A	N/A
13	99.95 ± 0.00	0.00 ± 0.00	92.85 ± 0.01	9.80 ± 0.01	N/A	N/A
14	99.95 ± 0.00	0.00 ± 0.00	92.95 ± 0.01	9.66 ± 0.01	N/A	N/A
15	99.95 ± 0.00	0.00 ± 0.00	93.17 ± 0.01	9.36 ± 0.01	N/A	N/A
16	99.95 ± 0.00	0.00 ± 0.00	93.28 ± 0.01	9.21 ± 0.01	N/A	N/A
17	**99.97 ± 0.00**	**0.00 ± 0.00**	93.30 ± 0.01	9.17 ± 0.01	N/A	N/A
18	**99.97 ± 0.00**	**0.00 ± 0.00**	93.36 ± 0.01	9.10 ± 0.01	N/A	N/A
19-20	**99.97 ± 0.00**	**0.00 ± 0.00**	**93.49 ± 0.01**	**8.92 ± 0.01**	N/A	N/A

Table 4: Comparison of accuracy (A) and false positive rate (FP) of intrusion detectors generated by IM(n), NB n-gram, and SVM n-gram on the *UNM denial of service* data. The accuracies and false positive rates were estimated using 10-fold cross-validation. We calculated a 99% confidence interval on the accuracies and false positive rates. Note that when $n \geq 3$, SVM n-gram is infeasible because of computational and memory requirements.

N	IM(n)		NB n-gram		SVM n-gram	
	A	FP	A	FP	A	FP
1	98.69 ± 0.00	0.92 ± 0.00	98.69 ± 0.00	0.92 ± 0.00	99.98 ± 0.00	0.01 ± 0.00
2	99.27 ± 0.00	0.01 ± 0.00	99.24 ± 0.00	0.04 ± 0.00	**99.99 ± 0.00**	**0.00 ± 0.00**
3	99.24 ± 0.00	0.01 ± 0.00	99.06 ± 0.00	0.69 ± 0.00	N/A	N/A
4	99.23 ± 0.00	0.01 ± 0.00	99.18 ± 0.00	0.68 ± 0.00	N/A	N/A
5	99.23 ± 0.00	0.01 ± 0.00	99.24 ± 0.00	0.65 ± 0.00	N/A	N/A
6	99.23 ± 0.00	0.01 ± 0.00	99.32 ± 0.00	0.63 ± 0.00	N/A	N/A
7	99.24 ± 0.00	0.01 ± 0.00	99.40 ± 0.00	0.59 ± 0.00	N/A	N/A
8	99.25 ± 0.00	0.00 ± 0.00	99.52 ± 0.00	0.47 ± 0.00	N/A	N/A
9	99.36 ± 0.00	0.00 ± 0.00	99.66 ± 0.00	0.33 ± 0.00	N/A	N/A
10	99.65 ± 0.00	0.00 ± 0.00	99.66 ± 0.00	0.33 ± 0.00	N/A	N/A
11	99.80 ± 0.00	0.00 ± 0.00	99.66 ± 0.00	0.33 ± 0.00	N/A	N/A
12	99.84 ± 0.00	0.00 ± 0.00	99.66 ± 0.00	0.33 ± 0.00	N/A	N/A
13	99.87 ± 0.00	0.00 ± 0.00	99.66 ± 0.00	0.33 ± 0.00	N/A	N/A
14	99.88 ± 0.00	0.00 ± 0.00	99.67 ± 0.00	0.31 ± 0.00	N/A	N/A
15	99.89 ± 0.00	0.00 ± 0.00	99.67 ± 0.00	0.31 ± 0.00	N/A	N/A
16	99.89 ± 0.00	0.00 ± 0.00	99.68 ± 0.00	0.31 ± 0.00	N/A	N/A
17	99.91 ± 0.00	0.00 ± 0.00	99.69 ± 0.00	0.30 ± 0.00	N/A	N/A
18	99.92 ± 0.00	0.00 ± 0.00	99.69 ± 0.00	0.30 ± 0.00	N/A	N/A
19-20	**99.93 ± 0.00**	**0.00 ± 0.00**	**99.69 ± 0.00**	**0.30 ± 0.00**	N/A	N/A

Table 5: Comparison of accuracy (A) and false positive rate (FP) of intrusion detectors generated by IM(n), NB n-gram, and SVM n-gram on *MIT LL 1998* data. We followed the train/test setting specified in the data set description. Note that when $n \geq 4$, SVM n-gram is infeasible because of computational and memory requirements.

N	IM(n)		NB n-gram		SVM n-gram	
	A	FP	A	FP	A	FP
1	36.57	62.49	36.57	62.49	**81.55**	**7.45**
2	80.33	0.06	**80.36**	**0.03**	80.53	8.28
3	80.35	0.03	80.35	0.05	80.28	9.35
4	**80.37**	**0.00**	80.34	0.03	N/A	N/A
5	80.33	0.05	80.35	0.03	N/A	N/A
6	80.33	0.05	80.30	0.08	N/A	N/A
7	80.37	0.04	78.50	2.89	N/A	N/A
8	79.51	1.11	78.45	2.95	N/A	N/A
9	79.44	1.20	75.22	6.93	N/A	N/A
10	75.99	5.50	75.12	7.07	N/A	N/A
11	75.93	5.57	74.99	7.27	N/A	N/A
12	75.93	5.57	75.05	7.29	N/A	N/A
13	75.93	5.58	74.85	7.56	N/A	N/A
14	75.91	5.59	71.11	12.84	N/A	N/A
15	75.79	5.76	65.10	25.30	N/A	N/A
16	76.97	6.26	52.47	42.21	N/A	N/A
17	72.23	12.46	50.39	44.82	N/A	N/A
18	71.98	12.67	50.42	44.77	N/A	N/A
19	71.99	12.66	50.36	44.88	N/A	N/A
20	74.95	9.06	49.88	44.93	N/A	N/A

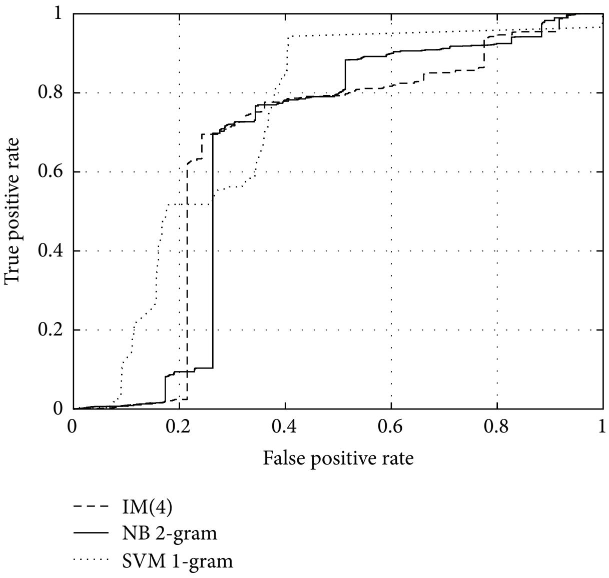


Figure 5: Receiver Operating Characteristic (ROC) curves of interelement dependency model of 4-gram (IM(4), AUC = 0.7290), Naive Bayes of 2-gram (NB 2-gram, AUC = 0.6654), and Support Vector Machines of 1-gram (SVM 1-gram, AUC = 0.7043) on the *MIT LL 1998* data estimated with the train/test setting.

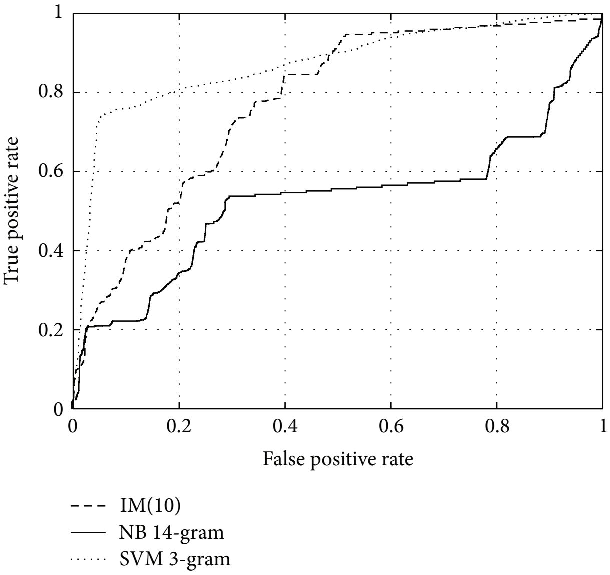


Figure 6: Receiver Operating Characteristic (ROC) curves of interelement dependency model of 10-gram (IM(10), AUC = 0.7742), Naive Bayes of 14-gram (NB 14-gram, AUC = 0.5271), and Support Vector Machines of 3-gram (SVM 3-gram, AUC = 0.8729) on *MIT LL 1998* data estimated using 10-fold cross-validation.

In Table 5, we show the results of training and testing experiments on the MIT LL 1998 data sets. In these training and testing experiments, we use predefined training data sets of the MIT LL 1998 data sets for learning intrusion detectors and use predefined testing data sets for evaluating the intrusion detectors. Notice that we do not calculate confidence intervals because it is infeasible to obtain distributions from the train/test setting.

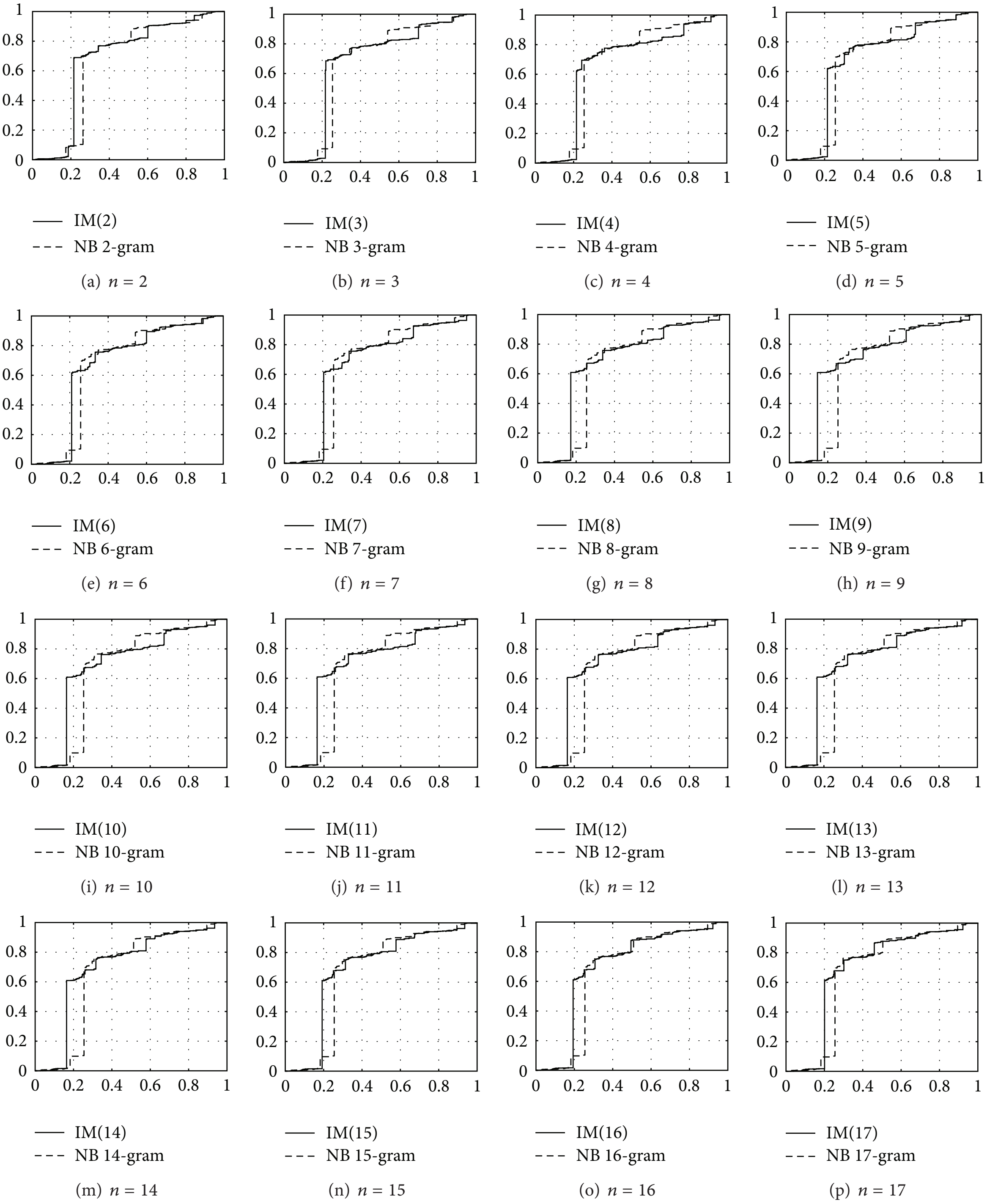
It can be seen that SVM with 1-gram shows the best performance on the accuracy with 81.55, but the false positive rate is 7.45. The difference from the results of IM(4) is marginal (accuracy is 80.37 and false positive rate is 0.00). Moreover, the false positive rate of IM(4) is far better than that of SVM with 1-gram. As discussed, the total learning and testing time of SVM is more than several hours, while, in contrast, it takes about one or two minutes for all the IM(n) models and their performances in general are comparable to those of SVMs.

In Figure 5, we show the Receiver Operating Characteristic (ROC) curves of train/test experiments on the MIT LL 1998 data sets. Area under the curve (AUC) of IM(4) is 0.7290, AUC of NB 2-gram is 0.6654, and AUC of SVM 1-gram is 0.7043. Thus, in terms of ROC, IM(4) outperforms SVM 1-gram.

In Table 6, we show the results of 10-fold cross-validation on the MIT LL 1998 data sets.

It can be seen that SVM with 3-grams shows the best performance with the accuracy 93.34 and the false positive rate 4.40. Again, the difference from the results of IM(n) is marginal (accuracy is 93.24 and false positive rate is 3.93). Moreover, the running time of SVM 3-grams is more than a week for learning and testing, whereas the running time of IM(n) is less than two minutes and the performance of IM(n) is comparable to that of SVM.

In Figure 6, we show the Receiver Operating Characteristic (ROC) curves of train/test experiments on the MIT LL



(a) $n = 2$ (b) $n = 3$ (c) $n = 4$ (d) $n = 5$

(e) $n = 6$ (f) $n = 7$ (g) $n = 8$ (h) $n = 9$

(i) $n = 10$ (j) $n = 11$ (k) $n = 12$ (l) $n = 13$

(m) $n = 14$ (n) $n = 15$ (o) $n = 16$ (p) $n = 17$

FIGURE 7: ROC curves of the train/test experiments on the MIT LL 1998 data sets, when $2 \leq n \leq 17$.

1998 data sets. Area Under the Curve (AUC) of IM(10) is 0.7742, AUC of NB 2-gram is 0.5271, and AUC of SVM 1-gram is 0.8729. Thus, in terms of ROC, SVM 1-gram shows the best performance among the three.

To show the big picture of comparing IM(n) and NB n-gram, in Figure 7, we show ROC curves of the train/test experiments on the MIT LL 1998 data sets, when $2 \leq n \leq 17$.

4. Conclusion

4.1. Related Work. Peng and Schuurmans [8] introduced n-gram augmented Naive Bayes and applied their algorithms to text classification. Silvescu et al. [10] proposed interelement dependency models which are similar to n-gram-augmented Naive Bayes. However, to the best of our knowledge, there has

Table 6: Comparison of accuracy (A) and false positive rate (FP) of intrusion detectors generated by IM(n), NB n-gram, and SVM n-gram on *MIT LL 1998* data. The accuracies and false positive rates were estimated using 10-fold cross-validation. We calculated a 99% confidence interval on the accuracies and false positive rates. Note that when $n \geq 3$, SVM n-gram is infeasible because of computational and memory requirements.

N	IM(n)		NB n-gram		SVM n-gram	
	A	FP	A	FP	A	FP
1	32.55 ± 0.37	88.55 ± 0.25	32.55 ± 0.37	88.55 ± 0.25	91.83 ± 0.21	0.03 ± 0.01
2	74.54 ± 0.34	26.17 ± 0.34	60.61 ± 0.38	41.91 ± 0.38	92.45 ± 0.21	4.46 ± 0.16
3	48.33 ± 0.39	56.18 ± 0.39	46.97 ± 0.39	57.44 ± 0.39	**93.34 ± 0.19**	**4.40 ± 0.16**
4	85.39 ± 0.28	13.84 ± 0.27	50.73 ± 0.39	53.29 ± 0.39	N/A	N/A
5	92.17 ± 0.21	6.02 ± 0.19	56.28 ± 0.39	47.08 ± 0.39	N/A	N/A
6	93.12 ± 0.20	4.97 ± 0.17	60.77 ± 0.38	42.02 ± 0.38	N/A	N/A
7	93.17 ± 0.20	3.97 ± 0.15	76.68 ± 0.33	24.54 ± 0.33	N/A	N/A
8	93.23 ± 0.20	3.96 ± 0.15	77.58 ± 0.33	22.43 ± 0.33	N/A	N/A
9	93.22 ± 0.20	3.94 ± 0.15	78.74 ± 0.32	21.18 ± 0.32	N/A	N/A
10	**93.24 ± 0.20**	**3.93 ± 0.15**	79.33 ± 0.32	20.532 ± 0.32	N/A	N/A
11	93.24 ± 0.20	3.94 ± 0.15	78.71 ± 0.32	21.92 ± 0.32	N/A	N/A
12	93.37 ± 0.19	4.00 ± 0.15	78.64 ± 0.32	21.54 ± 0.32	N/A	N/A
13	91.65 ± 0.22	1.44 ± 0.09	82.43 ± 0.30	12.55 ± 0.26	N/A	N/A
14	91.39 ± 0.22	1.73 ± 0.10	**86.21 ± 0.27**	**8.12 ± 0.21**	N/A	N/A
15	91.35 ± 0.22	1.77 ± 0.10	85.09 ± 0.28	14.54 ± 0.27	N/A	N/A
16	91.27 ± 0.22	2.36 ± 0.12	85.10 ± 0.28	13.92 ± 0.27	N/A	N/A
17	91.50 ± 0.22	5.39 ± 0.18	85.24 ± 0.28	13.20 ± 0.26	N/A	N/A
18	91.42 ± 0.22	4.98 ± 0.17	85.37 ± 0.28	13.03 ± 0.26	N/A	N/A
19	91.47 ± 0.22	5.41 ± 0.18	85.54 ± 0.27	12.83 ± 0.26	N/A	N/A
20	91.46 ± 0.22	5.43 ± 0.18	85.68 ± 0.27	13.87 ± 0.27	N/A	N/A

been no research on application of interelement dependency models for intrusion detection tasks.

Rieck and Laskov [4] used language models to detect unknown network attacks. They used a trie data structure [25, 26] to compute the similarity between two traces. The generalized suffix tree we used is more advantageous for storing n-gram features in that it does not take constant time to find a substring in a trie data structure. We performed our experiments with n ranging from 1 to 20, whereas Rieck and Laskov did the experiments when n is only 1, 3, and 5.

Most intrusion detection research focuses on the n-gram approach [4, 14, 18, 19]. However, there have been a few research efforts [15–17] that explore intrusion detection tasks with a bag or set of system calls. Liao and Vemuri [15] applied the K-Nearest Neighbor text classification method to intrusion detection tasks. Kang et al. [16] proposed a bag of system calls representation for intrusion detection. They performed various experiments for misuse detection and anomaly detection and showed that a bag of system calls representation is effective for misuse detection. Liu et al. [17] compared different system call feature representations and concluded that system call alone is not sufficient for detecting insider threats. Our work extends the representation to n-gram and shows that the resulting intrusion detectors are more accurate than n-gram features in terms of accuracy and false positive rate.

Forrest et al. [18] devised a Sequence Time-Delay Embedding (STIDE) intrusion detector which takes the n-gram approach with a few thresholds. Tan and Maxion [23] tried to define the operational limit of STIDE and concluded that 6-gram is enough for anomaly detection. Our work shows that the interelement dependency model (IM(n)) is more accurate than the n-gram approach and, for some attack types (UNM live lpr MIT and UNM denial of service), six is not always the magic number.

4.2. Summary and Future Work. We discussed an application of interelement dependency models to n-grams stored in a k-truncated generalized suffix tree (k-TGST) directly to classify intrusive sequences. We evaluated the performance of our method with those of Naive Bayes and Support Vector Machines (SVMs) with n-gram features by the experiments on intrusion detection benchmark data sets.

Experimental results on the University of New Mexico (UNM) benchmark data sets and MIT Lincoln Lab 1998 DARPA intrusion detection evaluation data sets show that our method, which solves the problem of independence violation that happens when n-gram-features are directly applied to Naive Bayes (i.e., Naive Bayes with n-gram features), yields intrusion detectors with higher accuracy than those from Naive Bayes with n-gram features and shows comparable accuracy to those from SVM with n-gram features.

For scalable and lightweight counting of n-gram features, we use the k-truncated generalized suffix tree mechanism for storing n-gram features. With this mechanism, we tested the performance of the classifiers up to 20-gram in our experiment, which illustrates the scalability and accuracy of n-gram augmented Naive Bayes with the k-truncated generalized suffix tree storage mechanism.

As for future work, we plan to apply n-gram representation to system call arguments [27], because system call arguments are important for accurate intrusion detection. Since there has been a lot of research on n-gram approaches, it is of interest to devise an n-gram representation where each element is the audit record data structure (i.e., system call and its arguments). However, there have been no researches we are aware of for the application of n-gram representation for the audit record data structure.

References

[1] E. Charniak, *Statistical Language Learning*, MIT Press, Cambridge, MA, USA, 1994.

[2] W. Lee, S. J. Stolfo, and K. W. Mok, "A data mining framework for building intrusion detection models," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 120–132, May 1999.

[3] A. Murali and M. Rao, "A survey on intrusion detection approaches," in *Proceedings of the 1st International Conference on Information and Communication Technology (ICICT '05)*, pp. 233–240, August 2005.

[4] K. Rieck and P. Laskov, "Detecting unknown network attacks using language models," in *Proceedings of the 3rd International Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA '06)*, pp. 74–90, Berlin, Germany, 2006.

[5] M. Z. Shafiq, S. A. Khayam, and M. Farooq, "Embedded malware detection using Markov n-grams," in *Proceedings of the 5th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA '08)*, pp. 88–107, Paris, France, 2008.

[6] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "Training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pp. 144–152, ACM Press, New York, NY, USA, July 1992.

[7] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

[8] F. Peng and D. Schuurmans, "Combining naive Bayes and n-gram language models for text classification," in *Advances in Information Retrieval*, F. Sebastiani, Ed., vol. 2633 of *Lecture Notes in Computer Science*, pp. 335–350, New York, NY, USA, 2003.

[9] C. Andorf, A. Silvescu, D. Dobbs, and V. Honavar, "Learning classifiers for assigning protein sequences to gene ontology functional families," in *Proceedings of the 5th International Conference on Knowledge Based Computer Systems (KBCS '04)*, pp. 256–265, 2004.

[10] A. Silvescu, C. Andorf, D. Dobbs, and V. Honavar, "Inter-element dependency models for sequence classification," Tech. Rep., Iowa State University, 2004.

[11] D. Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, Cambridge University Press, New York, NY, USA, 1st edition, 1997.

[12] J. C. Na and K. Park, "Data compression with truncated suffix trees," in *Proceedings of the Data Compression Conference (DDC '00)*, Snowbird, UT, USA, March 2000.

[13] T. M. Mitchell, *Machine Learning*, McGraw-Hill, New York, NY, USA, 1997.

[14] C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting intrusions using system calls: alternative data models," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 133–145, May 1999.

[15] Y. Liao and V. R. Vemuri, "Using text categorization techniques for intrusion detection," in *Proceedings of the 11th USENIX Security Symposium*, pp. 51–59, USENIX Association, Berkeley, CA, USA, 2002.

[16] D.-K. Kang, D. Fuller, and V. Honavar, "Learning classifiers for misuse and anomaly detection using a bag of system calls representation," in *Proceedings of the 6th Annual IEEE System, Man and Cybernetics Information Assurance Workshop (SMC '05)*, pp. 118–125, West Point, NY, USA, June 2005.

[17] A. Liu, C. Martin, T. Hetherington, and S. Matzner, "A comparison of system call feature representations for insider threat detection," in *Proceedings of the 6th Annual IEEE System, Man and Cybernetics Information Assurance Workshop (SMC '05)*, pp. 340–347, West Point, NY, USA, June 2005.

[18] S. Forrest, L. Allen, A. S. Perelson, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, pp. 202–212, May 1994.

[19] W. Lee and S. Stolfo, "Data mining approaches for intrusion detection," in *Proceedings of the 7th USENIX Security Symposium*, pp. 79–94, San Antonio, TX, USA, 1998.

[20] R. G. Cowell, S. L. Lauritzen, A. P. David, D. J. Spiegelhalter, and D. J. Spiegelhater, *Probabilistic Networks and Expert Systems*, Springer, Secaucus, NJ, USA, 1999.

[21] E. Ukkonen, "On-line construction of suffix trees," *Algorithmica*, vol. 14, no. 3, pp. 249–260, 1995.

[22] R. Lippmann, D. Fried, I. Graf et al., "Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation," in *Proceedings of the DARPA Information Survivability Conference and Exposition*, IEEE Computer Society Press, Los Alamitos, CA, USA, 2000.

[23] K. M. C. Tan and R. A. Maxion, "'Why 6?' Defining the operational limits of stide, an anomaly-based intrusion detector," in *Proceedings of the Symposium on Security and Privacy*, pp. 188–201, May 2002.

[24] Platt and J. C. :, "Fast training of support vector machines using sequential minimal optimization," *Advances in Kernel Methods*, pp. 185–208, 1999.

[25] R. de la Briandais, "File searching using variable length keys," in *Proceedings of the AFIPS West Joint Computer Conference*, vol. 15, pp. 295–298, 1959.

[26] E. Fredkin, "Trie memory," *Communications of the ACM*, vol. 3, no. 9, pp. 490–499, 1960.

[27] D. Mutz, F. Valeur, G. Vigna, and C. Kruegel, "Anomalous system call detection," *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp. 61–93, 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 304628, 9 pages
http://dx.doi.org/10.1155/2013/304628

Research Article

Localization Techniques in Wireless Sensor Networks

Nabil Ali Alrajeh,[1] Maryam Bashir,[2] and Bilal Shams[2]

[1] *Biomedical Technology Department, College of Applied Medical Sciences, King Saud University, Riyadh 11633, Saudi Arabia*
[2] *Institute of Information Technology, Kohat University of Science and Technology (KUST), Kohat 26000, Pakistan*

Correspondence should be addressed to Nabil Ali Alrajeh; nabil@ksu.edu.sa

Received 22 May 2013; Accepted 15 June 2013

Academic Editor: S. Khan

The important function of a sensor network is to collect and forward data to destination. It is very important to know about the location of collected data. This kind of information can be obtained using localization technique in wireless sensor networks (WSNs). Localization is a way to determine the location of sensor nodes. Localization of sensor nodes is an interesting research area, and many works have been done so far. It is highly desirable to design low-cost, scalable, and efficient localization mechanisms for WSNs. In this paper, we discuss sensor node architecture and its applications, different localization techniques, and few possible future research directions.

1. Introduction

In WSNs, sensor nodes are deployed in real world environment and determine some physical behaviors. WSNs have many research challenges. Sensors are tiny devices, low costing, and having low processing capabilities. WSNs applications attracted great interest of researchers in recent years [1]. WSNs are different from ad hoc and mobile networks in many ways. WSNs have different applications; therefore, the protocols designed for ad hoc networks do not suit WSNs [2]. Different applications of WSNs are the following: monitoring environmental aspects and physical phenomena like temperature, sound, and light, habitat monitoring, traffic control monitoring, patient healthcare monitoring, and underwater acoustic monitoring. WSNs have many research issues that affect design and performance of overall network such as hardware and operating system [3], medium access schemes [4], deployment [5], time synchronization [6], localization, middleware, wireless sensors and actors networks [7], transport layer, network layer, quality of service, and network security [8]. WSNs applications have opened challenging and innovative research areas in telecommunication world especially in last few years. Localization of nodes is very crucial to find and determine location of sensor node with the help of specialized algorithm. Localization is the process of finding the position of nodes [9] as data and information are useless if the nodes have no idea of their geographical positions. GPS (global positioning system) is the simplest method for localization of nodes, but it becomes very expensive if large number of nodes exists in a given network. Many algorithms have been proposed to solve the issue of localization; however, most of existing algorithms are application specific and most of the solutions are not suitable for wide range of WSNs [10, 11]. Ultrawide band techniques are suitable for indoor environment while acoustic transmission-based system requires extra hardware. Both are accurate techniques but expensive in terms of energy consumption and processing. Unlocalized nodes estimate their positions from anchor nodes beacon messages, which requires much power. Many algorithms have been proposed to minimize this communication cost. If one node estimates its wrong location, then this error propagates to overall network and further nodes; as a result, wrong information of anchor nodes location is propagated. To find the position of nodes is mainly based on distance between anchor node (with known location) and unlocalized node (with unknown location). Sensor nodes are used in industrial, environmental, military, and civil applications [12].

In this paper, we discuss sensor node localization schemes having different features used for different applications. Different algorithms of localization are used for static sensor nodes and mobile sensor nodes. The rest of the paper is organized as follows. Section 2 discusses components of sensor

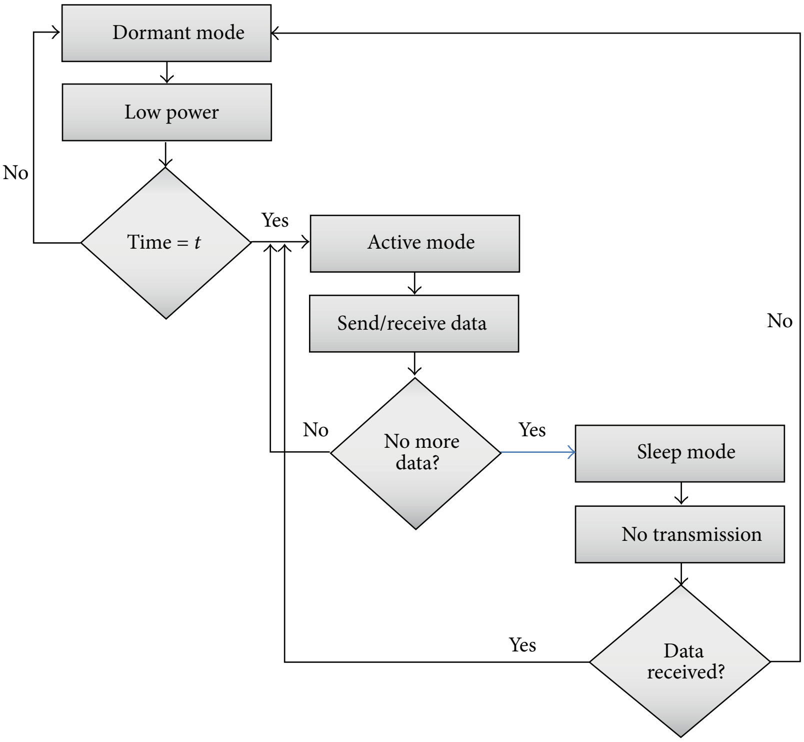


Figure 1: Transition of sensor node in different modes.

nodes. Section 3 describes WSNs applications. Section 4 provides overview of localization in WSNs. Section 5 presents range-free and range-based localization techniques. Section 6 covers analysis and discussion. Section 7 concludes the paper.

2. Components of Sensor Nodes

Sensor nodes have hardware and software components. Hardware components include processors, radio-transceiver sensors, and power unit. Softwares used for sensor nodes are Tinyos, Contiki, and Nano Rk. In this section, we discuss hardware components briefly.

2.1. Sensors. Sensors nodes are of two types: digital sensors and analog sensors. Analog sensors produce data in continuous or in wave form. The data is further processed by the processing unit that converts it to human readable form [12]. Digital sensors directly generate data in discrete or digital form. Once the data is converted, it directly sends it to the processor for further processing [12].

2.2. Processors. Microprocessors use different types of memory for processing data. The memory and input/output devices are integrated on same circuit. Random-access memory (RAM) stores data before sending it, while read-only memory (ROM) stores operating system of sensors nodes [13]. Microprocessors of sensor nodes are also known as tiny CPUs which are concerned about CPU speed, voltage, and power consumption. Sensors operations run at low CPU speed. Most of the time, sensors remain at sleep mode. When the processor is in sleep mode, this does not mean it is not consuming power. In sleep mode, it is involved in other activities like time synchronization [12].

2.3. Radio Transceiver. Transceiver receives and sends data to other sensor nodes [12]. It uses radio frequency to connect sensors with other nodes. Most of the energy is used by transceiver during data transmission. Transceiver has four operational modes such as sleep, idle, receive, and send [14].

2.3.1. Sleep Mode. In sleep mode, nodes turn off their communication devices or modules so that there is no more transmission and reception of data frames. In sleep mode, nodes can also listen to data frames. This is listening stage of sleep mode. When nodes listen to data frame, it shifts to active mode; otherwise, it remains in sleep mode.

2.3.2. Active Mode. In active mode, data is transmitted normally. Nodes communication devices are in active state and can send or receive data.

2.3.3. Dormant Mode. It is also one of the sleep modes. In this stage, sensor nodes are on low-power mode and remain in this mode for agreed amount of time. When sensor nodes go back to awake or active mode from dormant mode, they again rediscover networks and start communication [14].

The transition of sensor node in sleep, active, and dormant mode is presented in Figure 1.

2.4. Power Unit. It is the most important part of the sensor node. Sensor node cannot perform any work without this unit [13]. Power unit defines the lifetime of the sensor node.

Typical architecture of sensor node is given in Figure 2.

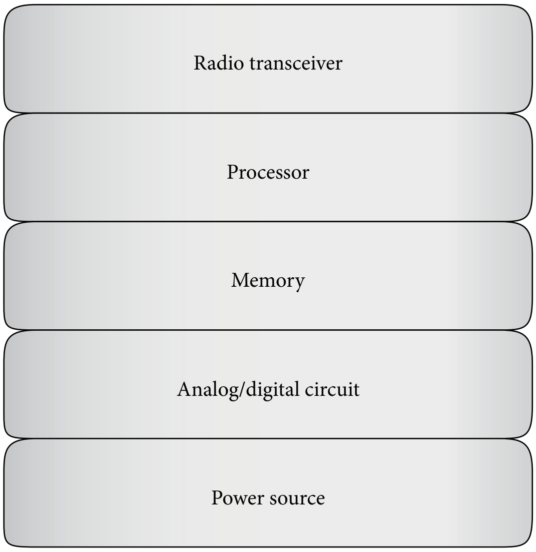


Figure 2: Typical architecture of sensor node.

3. Applications

Sensor nodes collect and forward data about particular application. Sensor nodes usually produce output when some kind of physical change occurs, such as change in temperature, sound, and pressure. WSNs have many applications such as military, civil, and environmental applications. Some important applications are discussed below.

3.1. Area Monitoring. Sensor nodes are deployed in the area where some actions have to be monitored; for instance, the position of the enemy is monitored by sensor nodes, and the information is sent to base station for further processing. Sensor nodes are also used to monitor vehicle movement.

3.2. Environmental Monitoring. WSNs have many applications in forests and oceans, and so forth. In forests, such networks are deployed for detecting fire. WSNs can detect when fire is started and how it is spreading. Senor nodes also detect the movements of animals to observe their habits. WSNs are also used to observe plants and soil movements.

3.3. Industrial Monitoring. In industries, sensors monitor the process of making goods. For instance, in manufacturing a vehicle, sensors detect whether the process is going right. A response is generated if there is any manufacturing fault [12]. Sensor nodes also monitor the grasping of objects by robots [12].

3.4. Medical and Healthcare Monitoring. Medical sensors are used to monitor the conditions of patients. Doctors can monitor patients' conditions, blood pressure, sugar level, and so forth, review ECG, and change drugs according to their conditions [12]. Personal health-monitoring sensors have special applications. Smart phones are used to monitor health, and response is generated if any health risk is detected. Medical sensors store health information and analyze the data obtained from many other sensors such as ECG, blood pressure, and blood sugar [15].

3.5. Traffic Control System. Sensor nodes monitor traffic flow and number plates of travelling vehicles and can locate their positions if needed. WSNs are used to monitor activities of drivers as well such as seat-belt monitoring [12].

3.6. Underwater Acoustic Sensor Networks. Underwater special sensors can monitor different applications of numerous oceanic phenomena; for instance, water pollution, underwater chemical reactions, and bioactivity. For such purposes, different types of 2D and 3D static sensors are used. 3D dynamic sensors are used to monitor autonomous underwater vehicles (AUVs) [12].

4. Localization Overview

Localization is estimated through communication between localized node and unlocalized node for determining their geometrical placement or position. Location is determined by means of distance and angle between nodes. There are many concepts used in localization such as the following.

(i) *Lateration* occurs when distance between nodes is measured to estimate location.

(ii) *Angulation* occurs when angle between nodes is measured to estimate location.

(iii) *Trilateration*. Location of node is estimated through distance measurement from three nodes. In this concept, intersection of three circles is calculated, which gives a single point which is a position of unlocalized node.

(iv) *Multilateration*. In this concept, more than three nodes are used in location estimation.

(v) *Triangulation*. In this mechanism, at least two angles of an unlocalized node from two localized nodes are measured to estimate its position. Trigonometric laws, law of sines and cosines are used to estimate node position [16].

Localization schemes are classified as anchor based or anchor free, centralized or distributed, GPS based or GPS free, fine grained or coarse grained, stationary or mobile sensor nodes, and range based or range free. We will briefly discuss all of these methods.

4.1. Anchor Based and Anchor Free. In anchor-based mechanisms, the positions of few nodes are known. Unlocalized nodes are localized by these known nodes positions. Accuracy is highly depending on the number of anchor nodes. Anchor-free algorithms estimate relative positions of nodes instead of computing absolute node positions [16].

4.2. Centralized and Distributed. In centralized schemes, all information is passed to one central point or node which is usually called "sink node or base station". Sink node computes position of nodes and forwards information to respected nodes. Computation cost of centralized based algorithm is decreased, and it takes less energy as compared with computation at individual node. In distributed schemes, sensors calculate and estimate their positions individually and directly communicate with anchor nodes. There is no clustering in distributed schemes, and every node estimates its own position [17–20].

4.3. GPS Based and GPS Free. GPS-based schemes are very costly because GPS receiver has to be put on every node. Localization accuracy is very high as well. GPS-free algorithms do not use GPS, and they calculate the distance between the nodes relative to local network and are less costly as compared with GPS-based schemes [21, 22]. Some nodes need to be localized through GPS which are called anchor or beacon nodes that initiate the localization process [16].

4.4. Coarse Grained and Fine Grained. Fine-grained localization schemes result when localization methods use features of received signal strength, while coarse-grained localization schemes result without using received signal strength.

4.5. Stationary and Mobile Sensor Nodes. Localization algorithms are also designed according to field of sensor nodes in which they are deployed. Some nodes are static in nature and are fixed at one place, and the majority applications use static nodes. That is why many localization algorithms are designed for static nodes. Few applications use mobile sensor nodes, for which few mechanisms are designed [23].

5. Range-Free and Range-Based Localization

Range-based and range-free techniques are discussed deeply in this section.

5.1. Range-Free Methods. Range-free methods are distance vector (DV) hop, hop terrain, centroid system, APIT, and gradient algorithm. Range-free methods use radio connectivity to communicate between nodes to infer their location. In range-free schemes, distance measurement, angle of arrival, and special hardware are not used [24, 25].

5.1.1. DV Hop. DV hop estimates range between nodes using hop count. At least three anchor nodes broadcast coordinates with hop count across the network. The information propagates across the network from neighbor to neighbor node. When neighbor node receives such information, hop count is incremented by one [24]. In this way, unlocalized node can find number of hops away from anchor node [13]. All anchor nodes calculate shortest path from other nodes, and unlocalized nodes also calculate shortest path from all anchor nodes [26]. Average hop distance formula is calculated as follows: distance between two nodes/number of hops [13].

Unknown nodes use triangulation method to estimate their positions from three or more anchor nodes using hop count to measure shortest distance [26].

5.1.2. Hop Terrain. Hop terrain is similar to DV hop method in finding the distance between anchor node and unlocalized node. There are two parts in the method. In the first part, unlocalized node estimates its position from anchor node by using average hop distance formula which is *distance between two nodes/total number of hops.* This is initial position estimation. After initial position estimation, the second part executes, in which initial estimated position is broadcast to neighbor nodes. Neighbor nodes receive this information with distance information. A node refines its position until final position is met by using least square method [13].

5.1.3. Centroid System. Centroid system uses proximity-based grained localization algorithm that uses multiple anchor nodes, which broadcast their locations with (X_i, Y_i) coordinates. After receiving information, unlocalized nodes estimate their positions [24]. Anchor nodes are randomly deployed in the network area, and they localize themselves through GPS receiver [13]. Node localizes itself after receiving anchor node beacon signals using the following formula [24]:

$$\left(X_{\text{est}}, Y_{\text{est}}\right) = \left(\frac{X_i, + \cdots +, X_n}{N}, \frac{Y_i, + \cdots +, Y_n}{N}\right), \tag{1}$$

where X_{est} and Y_{est} are the estimated locations of unlocalized node.

5.1.4. APIT. In APIT (approximate point in triangulation) scheme, anchor nodes get location information from GPS or transmitters. Unlocalized node gets location information from overlapping triangles. The area is divided into overlapping triangles [13]. In APIT, the following four steps are included.

(i) Unlocalized nodes maintain table after receiving beacon messages from anchor nodes. The table contains information of anchor ID, location, and signal strength [13].

(ii) Unlocalized nodes select any three anchor nodes from area and check whether they are in triangle form. This test is called PIT (point in triangulation) test.

(iii) PIT test continue until accuracy of unlocalized node location is found by combination of any three anchor nodes.

(iv) At the end, center of gravity (COG) is calculated, which is intersection of all triangles where an unlocalized node is placed to find its estimated position [24].

5.1.5. Gradient Algorithm. In gradient algorithm, multilateration is used by unlocalized node to get its location. Gradient starts by anchor nodes and helps unlocalized nodes to estimate their positions from three anchor nodes by using multilateration [13]. It also uses hop count value which is initially set to 0 and incremented when it propagates to other neighboring nodes [13]. Every sensor node takes information of the shortest path from anchor nodes. Gradient algorithm follows fes steps such as the following:

(i) In the first step, anchor node broadcasts beacon message containing its coordinate and hop count value.

(ii) In the second step, unlocalized node calculates shortest path between itself and the anchor node from which it receives beacon signals [27]. To calculate estimated distance between anchor node and unlocalized node, the following mathematical equation is used [27]:

$$D_{ji} = h_{j,Ai} d_{\text{hop}}, \tag{2}$$

where d_{hop} is the estimated distance covered by one hop.

(iii) In the third step, error equation is used to get minimum error in which node calculates its coordinate by using multilateration [13] as follows:

$$E_j = \sum_{i=1}^{n} \left(d_{ji} - d^{ji} \right), \tag{3}$$

where d^{ji} is the estimated distance computed through gradient propagation.

5.2. Range-Based Localization. Range-based schemes are distance-estimation- and angle-estimation-based techniques. Important techniques used in range-based localization are received signal strength indication (RSSI), angle of arrival (AOA), time difference of arrival (TDOA), and time of arrival (TOA) [28–34].

5.2.1. Received Signal Strength Indication (RSSI). In RSSI, distance between transmitter and receiver is estimated by measuring signal strength at the receiver [16]. Propagation loss is also calculated, and it is converted into distance estimation. As the distance between transmitter and receiver is increased, power of signal strength is decreased. This is measured by RSSI using the following equation [13]:

$$P_r(d) = \frac{p_t G_t G_r \lambda 2}{(4\lambda)^2 d^2}, \tag{4}$$

where P_t = transmitted power, G_t = transmitter antenna gain, G_r = receiver antenna gain, and λ = wavelength of the transmitter signal in meters.

5.2.2. Angle of Arrival (AOA). Unlocalized node location can be estimated using angle of two anchors signals. These are the angles at which the anchors signals are received by the unlocalized nodes [16]. Unlocalized nodes use triangulation method to estimate their locations [13].

5.2.3. Time Difference of Arrival (TDOA). In this technique, the time difference of arrival radio and ultrasound signal is used. Each node is equipped with microphone and speaker [35]. Anchor node sends signals and waits for some fixed amount of time which is t_{delay}, then it generates *"chirps"* with the help of speaker. These signals are received by unlocalized node at t_{radio} time. When unlocalized node receives anchor's radio signals, it turns on microphone. When microphone detects chirps sent by anchor node, unlocalized node saves the time t_{sound} [35]. Unlocalized node uses this time information for calculating the distance between anchor and itself using the following equation [12]:

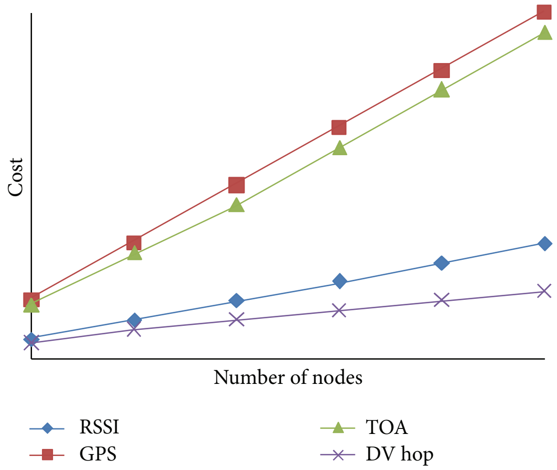


Figure 3: Cost analysis of localization techniques.

$$d = (s_{\text{radio}} - s_{\text{sound}}) * (t_{\text{sound}} - t_{\text{radio}} - t_{\text{delay}}). \tag{5}$$

5.2.4. Time of Arrival (TOA). In TOA, speed of wavelength and time of radio signals travelling between anchor node and unlocalized node is measured to estimate the location of unlocalized node [13]. GPS uses TOA, and it is a highly accurate technique; however, it requires high processing capability.

We generated some interesting results by comparing few localization techniques. The results are based on our observations and analysis.

Figure 3 shows cost of four localization techniques, and it is observed that GPS- and TOA-based systems are more expensive as compared with DV hop and RSSI.

Figure 4 represents accuracy comparison of different localization techniques. It is observed that localization mechanisms equipped with GPS systems are highly accurate.

Such mechanisms are needed for WSNs, which are energy efficient. Figure 5 shows comparison of energy efficiency of different localization mechanisms. GPS-based localization mechanisms are less energy efficient while RSSI-based mechanisms are highly energy efficient.

6. Analysis and Discussion

WSN has many constraints such as node size, energy, and cost. It is indeed necessary to consider these constraints before designing any localization mechanism. Nodes communication and data transmission take much power and consume more energy. Many localization algorithms have been proposed; however, most of them are application specific. Localization algorithm designed for one application is not

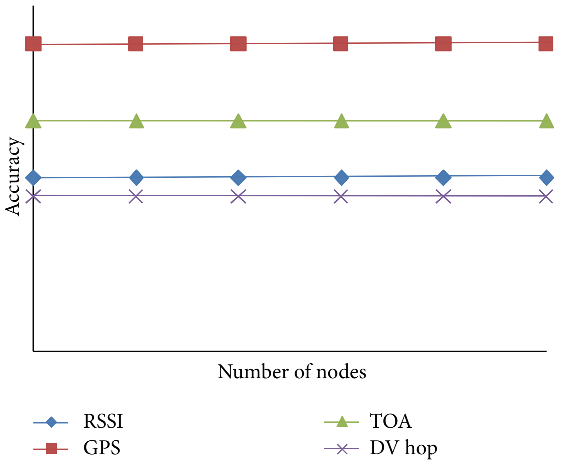


Figure 4: Accuracy comparison of different localization mechanisms.

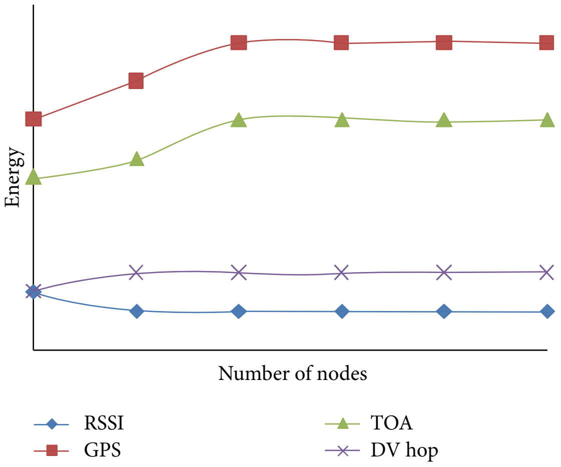


Figure 5: Energy efficiency comparison of different localization mechanisms.

necessarily suitable for other applications of WSNs. Similarly, some localization techniques are for mobile sensor nodes which are not suitable for static sensors.

When message travels from one node to another, the range is estimated by measuring time of flight of radio frequency. When time of flight of radio frequency is used with two messages, it is known as *two-way ranging (TWR) technique* [36]. In TWR mechanism, sensor network starts with at least three anchor nodes which are used as reference for unlocalized nodes [37–39]. Clock synchronization is not necessary because anchor and unlocalized nodes have their own reference clocks [40–42]. When unlocalized nodes become localized through trilateration method, they can localize other unlocalized nodes. The process continues until all unlocalized nodes become localized. To achieve high level of accuracy, nodes can refine their positions. Distance or range is estimated by measuring the round-trip time of signal transmission between anchor and unlocalized nodes [43]. TWR is suitable for WSNs because it is of low cost having less hardware requirements with less energy consumption. As mentioned above, TWR measures the round-trip time of signal transmission between two nodes. The emission of frames is triggered, and the process is controlled by MAC layer. The principle of TWR to calculate different time stamps between two nodes is presented in the following equation [44]:

$$D_{A,B} = v \cdot \frac{(t_4 - t_1) - (t_3 - t_2)}{2}, \qquad (6)$$

where v is the speed of light (radio signals), t_4 and t_2 are reception instants, and t_1 and t_3 are emission instants.

Clock synchronization is not a problem in this technique because each node has its own local clock and time stamps are calculated on these local clocks. When distance is measured by this formula, relatively accurate location can be estimated. Several authors proposed different protocols to analyze convergence speed and evaluate the communication cost related to energy. Different mechanisms are used to address some important issues such as the convergence conditions to achieve network wide localization, beginning and termination of the location propagation process, advantage of the broadcast nature of radio communications, and cooperation of nodes to reduce the number of exchanged messages and so energy consumption.

TWR has some issues such as clock drift in cheap sensor nodes, different response delays of nodes, and channel impairments.

In this section, we discuss the following three important protocols.

(i) *Beacon Protocol.* In beacon protocol, anchor node initiates the localization process. Localization wave originates from the center of the network and progresses towards the network boundaries. When this process stops, there is no way to provide localization facility to newly added nodes.

(ii) *Continuous Ranging Protocol.* In continuous ranging protocol, unlocalized node starts localization process, and its drawback is that unlocalized node sends range message even if localization wave still does not arrive.

(iii) *Optimized Beacon Protocol.* Optimized beacon protocol is the same as beacon protocol in which anchor nodes send range messages. However, when unlocalized node overhears three range messages, it starts localization process without waiting for neighbor node. When unlocalized node becomes localized node, it can localize other nodes as well.

Analysis shows that beacon protocol achieves the shortest delay and optimized beacon protocol requires fewer messages as compared with beacon and continuous ranging protocols. If estimation of location of node results in an error which is estimated by three neighbors' nodes, it decreases the accuracy of localization. The reason is that such error propagates for further localization process, so overall accuracy is compromised. Similarly, security of WSNs at various layers is a challenging task.

Localization of sensor nodes has numerous issues, and research community is trying to resolve them. In WSNs,

Table 1: Comparison of different localization techniques.

Technique	Cost	Accuracy	Energy efficient	Hardware size
GPS	High	High	Less	Large
GPS free	Low	Medium	Medium	Small
Centralized based	Depends	High	Less	Depends
Decentralized based	Depends	Low	High	Depends
RSSI	Low	Medium	High	Small
TOA (using ultrasonic pulse)	High	Medium	Less	Large
TDOA	Low	High	High	Less complex, may be large
AOA	High	Low	Medium	Large
DV hop	Low	Medium	High	Small
APIT	Medium	Medium	High	Medium

localization is an interesting research area, and still it has a lot of room for new researchers.

(i) During the designing of localization algorithm, designers must consider low cost of power, hardware, and deployment of localization algorithm.

(ii) Localization of sensor nodes using GPS is not suitable, because it is less energy efficient and expensive; it needs large size of hardware and has a line of sight problem. If GPS is installed on every node, then it increases the node size and deployment cost. Furthermore, GPS is not energy efficient as it consumes a lot of energy and not suitable for a network like WSN.

(iii) It is more challenging to design localization system for WSN as compared with other networks. As in WSNs, it is important to consider all of the limitations such as battery power, low processing, limited memory, low data rates, and small size.

(iv) In localization schemes, the problem of line of sight (LOS) can cause handling errors.

(v) Accuracy is highly important factor in all localization techniques. Localization accuracy is compromised when position of node is wrongly estimated. When a node localizes itself with wrong information of coordinates and propagates wrong information throughout the network, overall accuracy of the localization process is decreased.

(vi) Node density is an important factor in designing localization algorithm. For example, in beacon node-based algorithms, beacon node density should be high for accurate localization, whereas if node density is low, then accuracy is decreased and localization algorithms cannot perform well.

(vii) In mobile WSNs, nodes may leave and move to another location. In such case, topological changes may occur. In mobile WSNs, scalable mechanism is needed which is capable to cater topological changes. Furthermore, it is very difficult to estimate again and again position of node which is mobile in nature. Mobile nodes are continuously in motion.

(viii) *Fewer Number of Beacon Nodes* can be built means that localization accuracy is improved and accurate as number of beacon nodes increases, but localization algorithm should be accurate with fewer number of beacon nodes.

(ix) *3D Space*. Localization should be suitable for deployment in actual 3D space.

(x) *Universal Nature*. Algorithm should be designed not only for one specific application but for others as well. If algorithm is designed for indoor environments, it should also work for outdoor environments.

Table 1 presents comparison of different localization techniques.

7. Conclusions

WSNs have many applications in which sensor nodes collect data from particular location and process it. However, it is an important task to know the location of data from where it is collected. Localization is a mechanism in which nodes are located. There are many approaches for localization; however, such approaches are desirable which are capable to take care of limited resources of sensor nodes. In this paper, we explained different localization techniques in detail. Range-free, range-based, and TWR techniques are deeply analyzed.

Acknowledgments

The authors extend their appreciation to the Research Centre, College of Applied Medical Sciences and to the Deanship of the Scientific Research at King Saud University for funding this paper.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2005.

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

[3] S. Gowrishankar, T. G. Basavaraju, D. H. Manjaiah, and S. K. Sarkar, "Issues in wireless sensor networks," in *Proceeding of the World Congress on Engineering (WCE '08)*, vol. 1, London, UK, July 2008.

[4] N. Correal and N. Patwari, *Wireless sensor network: challenges and opportunities*, Florida Communication Research Lab, Gainesville, Fla, USA, 2001.

[5] D. Ganesan, A. Cerpa, W. Ye, Y. Yu, J. Zhao, and D. Estrin, "Networking issues in wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 799–814, 2004.

[6] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.

[7] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, no. 4, pp. 351–367, 2004.

[8] J. A. Stankovic, "Research challenges for wireless sensor networks," *ACM SIGBED Review*, vol. 1, no. 2, pp. 9–12, 2004.

[9] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[10] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom '01)*, pp. 166–179, Rome, Italy, July 2001.

[11] A. Savvides, H. Park, and M. B. Srivastava, "The n-hop multilateration primitive for node localization problems," *Mobile Networks and Applications*, vol. 8, no. 4, pp. 443–451, 2003.

[12] F. Hu and X. Cao, *Wireless Sensor Networks: Principles and Practice*, Auerbach, Boca Raton, Fla, USA, 1st edition, 2010.

[13] R. Manzoor, *Energy efficient localization in wireless sensor networks using noisy measurements [M.S. thesis]*, 2010.

[14] L. E. W. Van Hoesel, L. Dal Pont, and P. J. M. Havinga, *Design of an Autonomous Decentralized MAC Protocol for Wireless Sensor Networks*, Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands.

[15] M. A. Perillo and W. B. Heinzelman, "Wireless sensor network protocols," 2004.

[16] A. Youssef and M. Youssef, "A taxonomy of localization schemes for wireless sensor networks," in *Proceedings of the International Conference on Wireless Networks (ICWN '07)*, pp. 444–450, Las Vegas, Nev, USA, 2007.

[17] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Computer Networks*, vol. 43, no. 4, pp. 499–518, 2003.

[18] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 50–61, November 2004.

[19] C. Savarese, J. M. Rabaey, and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *Proceedings of the 2002 USENIX Annual Technical Conference on General Track*, pp. 317–327, USENIX Association, Berkeley, Calif, USA, 2002.

[20] J. Liu, Y. Zhang, and F. Zao, "Robusr distributed node localization with error management," in *Proceeding of the 7th ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc '06)*, pp. 250–261, Florence, Italy, May 2006.

[21] S. Qureshi, A. Asar, A. Rehman, and A. Baseer, "Swarm intelligence based detection of malicious beacon node for secure localization in wireless sensor networks," *Journal of Emerging Trends in Engineering and Applied Sciences*, vol. 2, no. 4, pp. 664–672, 2011.

[22] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, 2000.

[23] E. Kim and K. Kim, "Distance estimation with weighted least squares for mobile beacon-based localization in wireless sensor networks," *IEEE Signal Processing Letters*, vol. 7, no. 6, pp. 559–562, 2010.

[24] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th ACM Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 81–95, September 2003.

[25] T. He, C. Huang, B. M. Blum, J. A. Stankvic, and T. Abdelzaher, "Range free localization," 2006.

[26] Q. Huang and S. Selvakennedy, "A range-free localization algorithm for wireless sensor networks," in *Proceedings of the IEEE 63rd Vehicular Technology Conference (VTC '06)*, pp. 349–353, School of InformationTechnologies, Melbourne, Australia, July 2006.

[27] R. Stoleru, T. He, and J. A. Stankovic, "Range free localization," in *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*, vol. 30, pp. 3–31, Springer, Berlin, Germany, 2007.

[28] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, pp. 1655–1663, April 2001.

[29] A. Savvides, C. Han, and M. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the 7th ACM Annual International Conference on Mobile Computing and Networking (MobiCom '01)*, pp. 166–179, Rome, Italy, July 2001.

[30] A. Savvides, H. Park, and M. B. Srivastava, "The bits and flops of the n-hop multilateration primitive for node localization problems," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pp. 112–121, September 2002.

[31] A. Nasipuri and K. Li, "A directionality based location discovery scheme for wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pp. 105–111, September 2002.

[32] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2137–2148, 2003.

[33] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 1734–1743, San Francisco, Calif, USA, April 2003.

[34] D. Liu, P. Ning, and W. Du, "Detecting malicious beacon nodes for secure location discovery in wireless sensor networks," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems ((ICDCS '05))*, pp. 609–619, June 2005.

[35] J. Bachrach and C. Taylor, *Localization in Sensor Networks*, Massachusetts Institute of Technology, Cambridge, Mass, USA, 2004.

[36] S. Tanvir, *Energy efficient localization for wireless sensor network [Ph.D. thesis]*, 2010.

[37] R. Hach, "Symmetric double sided two-way ranging," IEEE 802. 15. 4a, Ranging Subcommittee, 2005.

[38] L. J. Xing, L. Zhiwei, and F. C. P. Shin, "Symmetric double side two way rangingwith unequal reply time," in *Proceedings of the IEEE 66th Vehicular Technology Conference (VTC '07)*, pp. 1980–1983, Baltimore, Md, USA, October 2007.

[39] J. Decuir, "Two way time transfer based ranging," Contribution to the IEEE 802. 15. 4a, Ranging Subcommittee, 2004.

[40] R. Manzoor, *Energy efficient localization in wireless sensor networks using noisy measurements [M.S. thesis]*, 2010.

[41] R. Hach, "Symmetric double sided two-way ranging," IEEE 802. 15. 4a, Ranging Subcommittee, 2005.

[42] N. Hajlaoui, I. Jabri, and M. B. Jemaa, "Experimental performance evaluation and frame aggregation enhancement in IEEE 802.11n WLANs," *International Journal of Communication Networks and Information Security*, vol. 5, no. 1, pp. 48–58, 2013.

[43] S. Tanvir, E. Schiller, B. Ponsard, and A. Duda, "Propagation protocols for network-wide localization based on two-way ranging," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '10)*, Sydney, Australia, April 2010.

[44] S. Tanvir, F. Jabeen, M. I. Khan, and B. Ponsard, "On propagation properties of beacon based localization protocol for wireless sensor networks," *Middle-East Journal of Scientific Research*, vol. 12, no. 2, pp. 131–140, 2012.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 326295, 9 pages
http://dx.doi.org/10.1155/2013/326295

Research Article

Secure Ant-Based Routing Protocol for Wireless Sensor Network

Nabil Ali Alrajeh, Mohamad Souheil Alabed, and Mohamed Shaaban Elwahiby

Biomedical Technology Department College of Applied Medical Sciences, King Saud University, Riyadh 11633, Saudi Arabia

Correspondence should be addressed to Nabil Ali Alrajeh; nabil@ksu.edu.sa

Received 25 May 2013; Accepted 16 June 2013

Academic Editor: S. Khan

Optimal path selection in wireless sensor networks (WSNs) is one of the challenging tasks. Several efficient routing protocols are proposed for specific scenarios to achieve particular objectives in WSN. However, such networks have many limitations such as low data rates and security threats. In this paper, we propose an adaptive secure routing protocol which is based on bioinspired mechanism. It uses distributed ant-based methodology to select two optimal paths keeping in view route security. Simulation results show that our routing protocol can perform better in many scenarios.

1. Introduction

WSNs are usually deployed in such areas where other wired and wireless networks are not feasible to be established and configured. It consists of low power sensor nodes and high power sink and is typically used for environmental monitoring. Sensor networks normally operate in decentralized and distributed manner in which sensor nodes have the capability of self-healing and self-configuration. Due to physical hostile environment and multihop and distributed architecture, WSN is more vulnerable to different types of security attacks. An attacker can easily launch security attacks against physical, media access, or network layer to WSN.

Security is one of the most important aspects in deploying or designing a sensor network. As WSN is deployed in harsh and extreme environment, it is not possible to protect it from security attacks by physical monitoring.

Most applications of WSN need security considerations especially military-based applications and monitoring. Sensor nodes are of low cost and small in size due to which heaving security mechanisms cannot be used. The reason is that heavy security mechanisms demand more processing power, more memory, and more battery resources which may increase the cost. Furthermore, most of sensor nodes are not tamper resistant, and the attacker can extract the sensitive data from the nodes and even configure few nodes for malicious activities and false data routing from source to destination.

Secure routing is one of the optimal solutions to counter network layer security attacks. In WSN, the data is more vulnerable to different security attacks during data transmission from source to multihop away destination [1, 2]. This is the reason; secure routing is always desirable in such kind of networks.

An attacker can conduct variety of security attacks against WSN, such as blackhole, greyhole, sinkhole, false routing updates, packet modification attack, packet misdirecting attack, and hello flood attack [3–7]. These security attacks bring serious routing malfunctions in data transmission from source to destination. Some attacks are less severe while some have more severity. For example, greyhole attack selectively forwards packets to the next hop, whereas blackhole and sinkhole attacks drop all the packets and create a denial of service (DoS) situation. These network layer security attacks can be prevented by appropriate secure routing protocol.

Secure routing is an important step for designing and deploying multihop wireless networks such as WSN. Multihop wireless networks are more vulnerable to security attacks as compared to single-hop wireless networks. The reason is that most of multihop wireless networks are distributed having no centralized body. Designing an appropriate secure routing protocol for WSN is a challenging task. In WSN the ideal routing protocol should be secure and efficient in terms of data delivery, route discovery, and routing overheads.

Research community has proposed many methodologies for designing secure routing protocols. Few important methodologies are cross-layered mechanisms, multipath mechanisms, and bioinspired mechanisms.

In cross-layer mechanism, parameters are exchanged across different layers of protocol suit for optimal path selection from source to destination [8, 9]. However, this methodology requires more computation, memory, and battery resources. The advantage of this methodology is its capability to counter multilayer security threats. In multipath mechanism, two or more paths are established from source to destination [10, 11]. The data is routed through many paths instead of one, and therefore such mechanisms are considered more fault tolerant as compared to single path. Bioinspired mechanisms are considered more robust as they provide interesting solution for routing due to their inherent scalable features [12].

In this paper, we present an adaptive secure routing protocol which is based on bioinspired technique termed as ant colonization. Our protocol has three important features which are listed as follows.

(i) *Adaptive Security Nature*. The proposed protocol provides adaptive security mechanism.

(ii) *Two Paths Selection*. The proposed routing mechanism selects two paths for data transmission. The proposed mechanism ensures security of both routes.

(iii) *Bioinspired Technique*. The proposed routing algorithm is based on ant colony optimization (ACO) technique.

Our proposal is able to select optimal paths from source to destination by ensuring adaptability, robustness, and security. The rest of the paper is organized as follows. Section 2 discusses related work. Protocol design considerations are covered in Section 3. Section 4 describes the evaluation and simulation results. Section 5 concludes the paper.

2. Related Work

WSNs have many real-life applications such as military applications, healthcare applications, forest and habitat monitoring, fire, heat, and pressure monitoring in a given area [13]. Selection and delivery of data packets from source to destination is one of the important tasks in WSN. Researchers have proposed many routing mechanisms so far [14–18]. However, majority of WSN routing protocols are application dependent without any consideration for security aspect. Security concern is gaining significant attention and many secure mechanisms have been proposed for WSN [19–23]. However, many of these security mechanisms operate at different layers to counter specific risk situation. To counter network layer security attacks, secure routing is more appropriate security mechanism. In last few years, variety of secure routing protocols are proposed for sensor networks [24–27].

In WSN, base station periodically broadcasts the routing information. The attacker can easily misdirect, drop, or modify the data during transit. A security mechanism termed as μTesla [28] is used to protect base station broadcast data from modifications. μTesla uses symmetric cryptography which provides authenticated broadcast. This scheme is expensive due to long one way hash chain. An authenticated routing message in sensor network (ARMS) [29] is proposed to overcome shortcomings of μTesla. It uses shared secret key and one way short hash chain. Enhancement of fault tolerant ad hoc on-demand distance vector (ENFAT-AODV) [30] is proposed to address the issue of node failure. A backup path is established which is used whenever the main route is not available due to node failure. Sensor nodes are deployed in distributed manner, so trusted neighbor discovery is an important task. Cross-layer secure routing protocol using energy harvesting mechanism in wireless sensor networks is proposed in [31]. Secure alternate path routing in sensor network (SeRINS) uses key management scheme along with neighbor report system [32]. The objective of this mechanism is to protect WSN from attacks such as packet modification or bogus routing information. Another secure approach that uses secret key cryptography with rekeying support is proposed for WSN [33].

As we know WSNs have resource constraints in terms of data rates, battery power, computation, and memory, and therefore it is highly desirable to design and propose such security mechanisms especially routing protocols which are light weighted so that the critical resources are reserved and to enable sensor nodes for long-time operations. Secret keys sharing and management requires more resources. Bioinspired mechanisms offer robust, fast, and inexpensive solutions for securing WSN. Ant colony optimization (ACO) is one of the bioinspired techniques, which provides robust and interesting solutions for WSN routing protocols.

Termite algorithm is inspired from the termite colonies [34]. Termite is a hybrid protocol in which route is discovered on demand by the ants while the data packets implicitly maintain the quality of paths in proactive manner. In this approach forward ants unicast and follow a random path while backward ants may or may not use the same path. Optimized termite [35] uses the concept of termite with the objective of enhancing load balancing mechanism. This algorithm selects such a route having less traffic. Ant-dymo [36] is basically proposed for ad hoc networks, and it is based on ACO. Ant-dymo aims to improve the end to end delay and packet loss. This algorithm uses two types of artificial ants, explorer ant, which explores the route in proactive manner, and search ant responsible for searching for specific destination in case it is not present in the routing table. AntHocNet [37] sends reactive forward ants for route discovery between source and destination and proactive forward ants are used to test the quality of existing paths and explore alternate best paths. AntOR [38] is based on Ant Colony Optimization with some modifications in AntHocNet routing algorithm. AntOR provides better load balancing by satisfying multiple quality of services constraints and lower control overhead by imposing some restriction on the exchange of routing information. GrAnt [39] is a prediction-based routing algorithm which provides higher packet delivery with low routing overhead. GrAnt depends on both local information and global information. In [40], ant-based framework for routing in WSN along with mathematical theory of bioinspired computation is presented. An energy efficient ant-colonization-based routing algorithm is proposed in [41]. In this algorithm,

Table 1: Different stages of proposed protocol.

Stage	Operation
Route discovery	Initial discovery of two or more routes
Route selection	Selection of appropriate routes
Security	Implementation of security mechanism
Data forwarding	Forwarding of data from source to destination

each individual packet is treated as an ant, which communicates with others through pheromone. Each sensor node maintains a table known as pheromone table. Biologically inspired optimization for sensor lifetime (Bio4sel) [42] is a decentralized, autonomic, and distributed ant-based routing algorithm that aims to increase sensor network lifetime. Another energy efficient routing protocol is presented in [43]. This scheme uses dynamic route identification scheme in case of path failure due to dead node or intrusion. The objective of this routing protocol is energy efficiency and path reliability.

Most of the bioinspired routing protocols based on ant optimization technique are efficient in terms of optimal path selections. ACO-based routing protocols are less expensive in terms of battery, computations, and memory usage; however, they cannot perform well in case of security attacks due to lack of security mechanism. As we know, WSNs are deployed in harsh and hostile environment, and therefore some sort of security mechanism is indeed necessary along with bioinspired methodology.

3. Proposed Routing Scheme

WSN is gaining more attention due to its wide range of usage and applications. Security consideration is one of the important factors as most of WSNs are deployed in harsh and hostile environment. That is why secure routing is highly desirable for transmitting data from source to destination. The proposed mechanism is secure routing based on ant colony optimization (ACO) technique.

The proposed routing protocol consists of four stages which are listed as follows:

(i) route discovery;

(ii) route selection;

(iii) route security;

(iv) data forwarding.

The proposed scheme is presented in Table 1.

Ants are able to select the shortest path from source to destination, that is, from home to a food. Ants drop a chemical substance termed as pheromone along the path. Generally, shortest path has more intensity and concentration of pheromone as compared to other paths. There are two important categories of ants; one is forward ants and the second is backward ants. In our scheme we utilize this scheme for route request and route reply mechanisms.

Before route discovery process, the proposed routing protocol uses simple mechanism to discover surrounding neighbors using hello packets. The routing table is updated on the basis of responses received from surrounding neighbors. During route discovery process, source node broadcasts a fixed number of ants (route request packets (RREQ)) to random neighbors. The source node also sets a lifetime interval of RREQ ants. These RREQ ants are acting as forward ants. Forward ants travel through intermediate nodes and find destination (food). The destination receives many RREQ ants through random intermediate nodes. However, total numbers of RREQ ants received at destination are equal to or less than that sent by source node, as some of the RREQ ants may not reach to destination due to congestion, expiration of lifetime interval, or path failure. The destination node sends back route reply (RREP) ants using the same paths which are used by forward ants. RREP ants act as backward ant. The packet formats of forward and backward ants are almost same; however, backward ant has a field *path_security*, instead of *reputation_value*. Path security field accepts only two values; that is, 1 or 0. 0 means that the path is not secure and needs security mechanism while 1 means that the path is secure and there is no need to enable security mechanism.

Table 2: Packet format of forward ant.

Field	Description
S_ID	Source ID of originating node
D_ID	Destination ID
H_count	Hop count from source to destination
Lifetime	Lifetime of packet
Reputation_value	Trust and reputation value of next neighbor

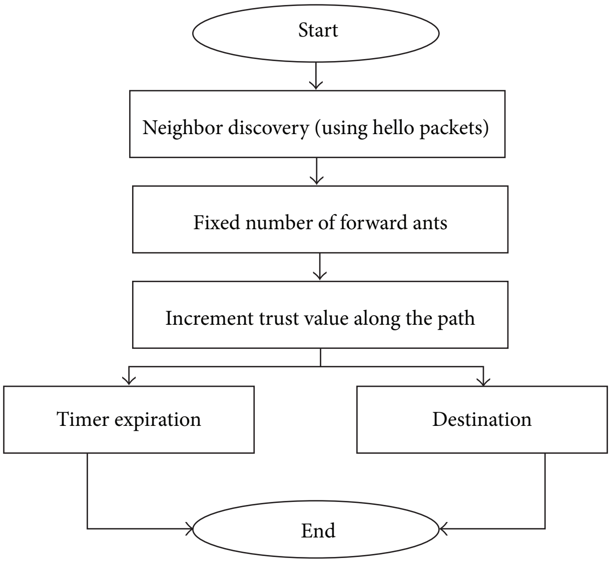

Start
Neighbor discovery (using hello packets)
Fixed number of forward ants
Increment trust value along the path
Timer expiration
Destination
End

Figure 1: Forward ant process.

The packet format of forward ant is given in Table 2.

The algorithm for forward ant process is presented in Figure 1.

The algorithm for backward ant process is presented in Figure 2.

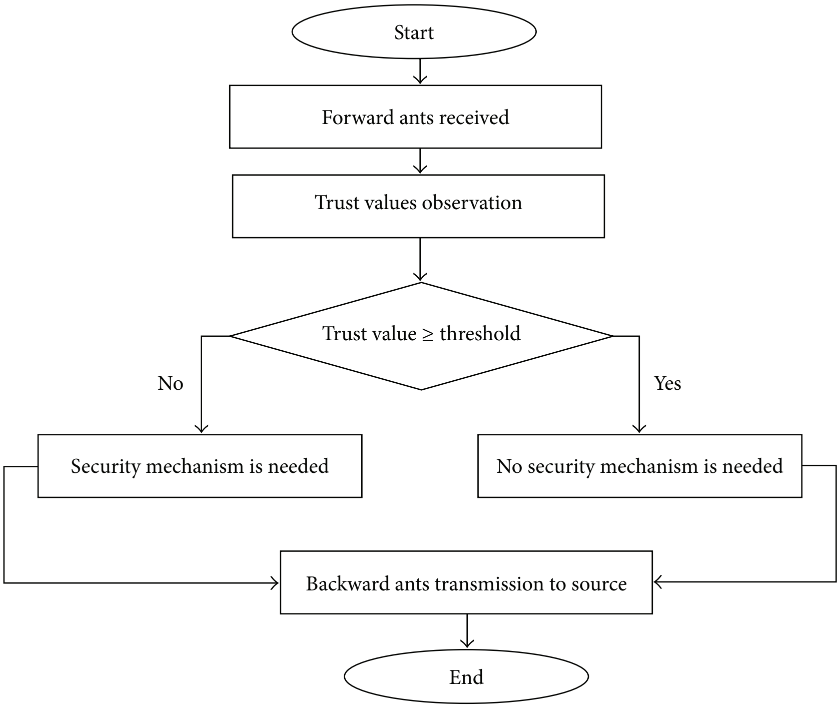

Start
Forward ants received
Trust values observation
Trust value ≥ threshold
No
Yes
Security mechanism is needed
No security mechanism is needed
Backward ants transmission to source
End

Figure 2: Backward ant process.

The route discovery processes have two important features.

(i) Propagation of fixed numbers of forward ants, which are responsible to increment reputation value along multihop path till destination, is clear.

(ii) Destination node receives random numbers of forward ants and responds using backward ant propagation towards source node. The destination node enables *path_security* field to specify whether the source node needs to ensure security or not by observing reputation values.

When source node receives many backward ants RREPs packets, then route selection procedure is triggered. In route selection process, source node examines all the received backward ants and chooses two paths considering security features. The route selection process is presented in Figure 3.

The route discovery and route selection process of proposed mechanisms is inspired from ACO [44]. Suppose that there are four ants (A1, A2, A3, and A4) and four different routes (R1, R2, R3, and R4) are available to food (F). The four ants at the starting point (S1) have no knowledge of the food.

(i) Ant A1 selects route R1. Similarly ants A2, A3, and A4 select routes R2, R3, and R4, respectively.

(ii) A1, A2, A3, and A4 randomly reach food (F).

(iii) Suppose that Ant A3 faces less hurdles along the path, so it left high concentration of pheromone. It means that the trust value of R3 is greater as compared to other routes.

(iv) At destination, the trust values of all paths are computed.

(v) From destination, all four ants followed their own path toward the starting point.

(vi) At starting point, two paths are now selected on the basis of path_security value.

(vii) Now the ants travel using those two paths which are more secure.

The security mechanism in proposed routing protocol is based on watchdog mechanism [45]. In our scheme, watchdog is implemented on every sensor node. Sensor nodes monitor all surrounding neighbors in their radio range using watchdog. On the basis of the mutual monitoring mechanisms, node can classify its neighbors as cooperative or noncooperative. Watchdog mechanism is capable to monitor neighbors, collects data, and observe data forwarding behavior of surrounding nodes. In our approach, a reputation value is assigned to all neighbors, and whenever a forward ant arrives, the reputation value of next neighbor is incremented.

The destination node (sink) makes the decision by enabling 0 or 1 in the backward ant propagation whether source node needs to send encrypted data or not. When source node receives the backward ant having 0 in the path_security field, an encryption mechanism is used to ensure data security. We are using lightweight encryption mechanism which is discussed in [46]. In this mechanism, permutated key is generated by RC4. The key to RC4 encryption changes dynamically, and thus every data packet has a different dynamically generated key.

The complete working mechanism of proposed routing protocol is presented in Figure 4.

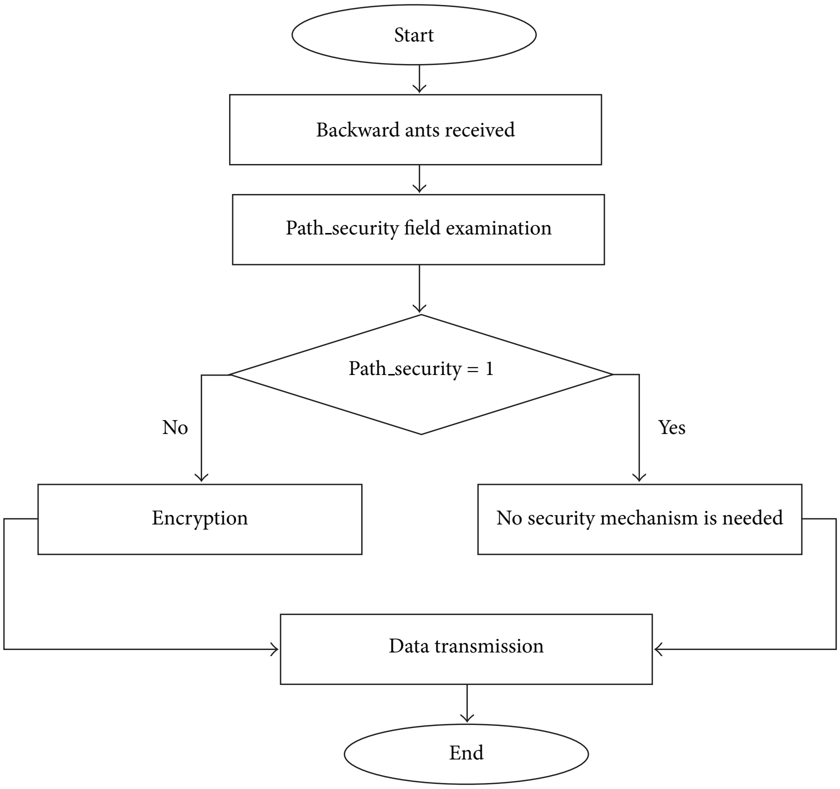


FIGURE 3: Route selection process.

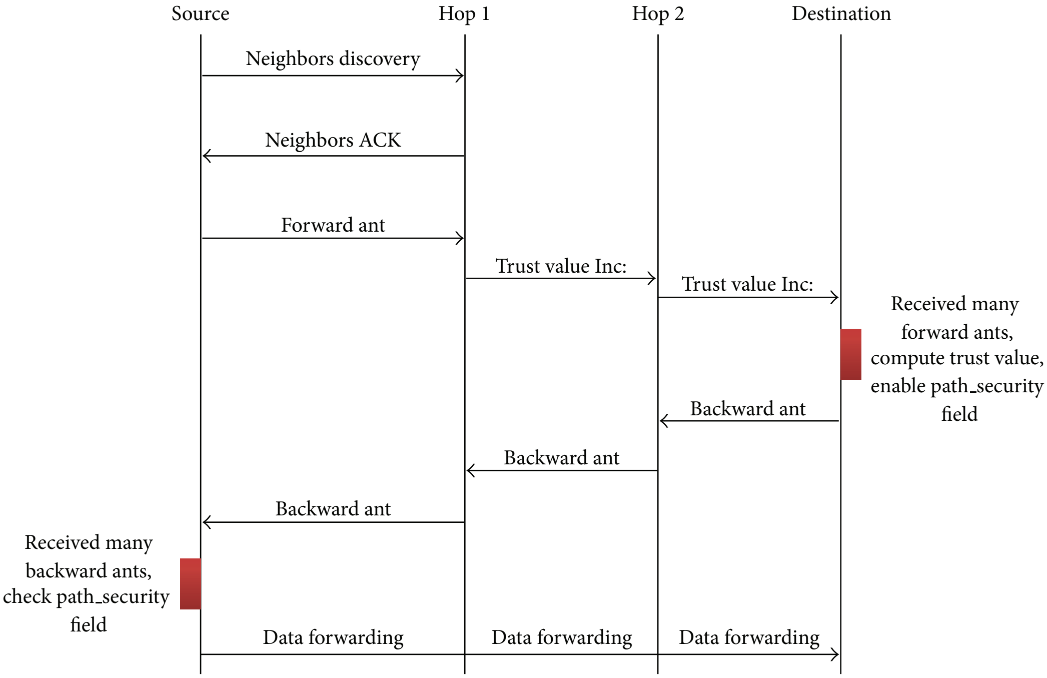


FIGURE 4: Complete working mechanism of proposed routing protocol.

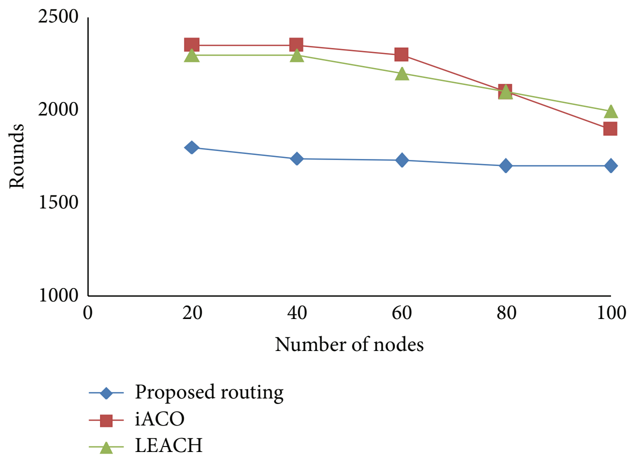


Figure 5: Network life time based on number of rounds.

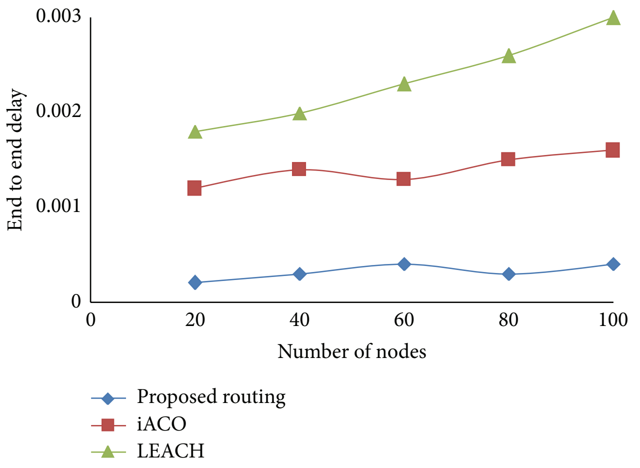


Figure 7: End to end delay.

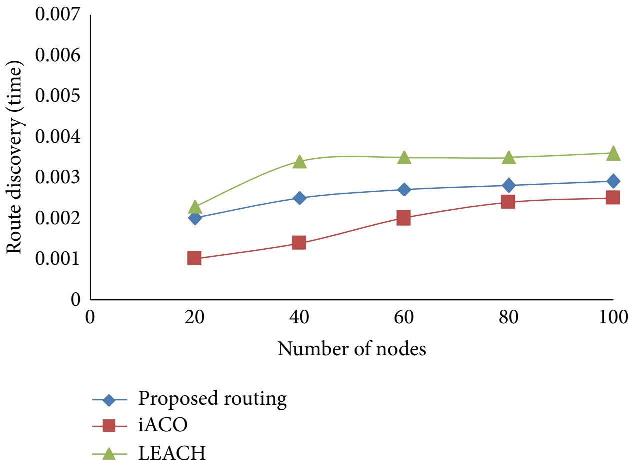


Figure 6: Route discovery efficiency.

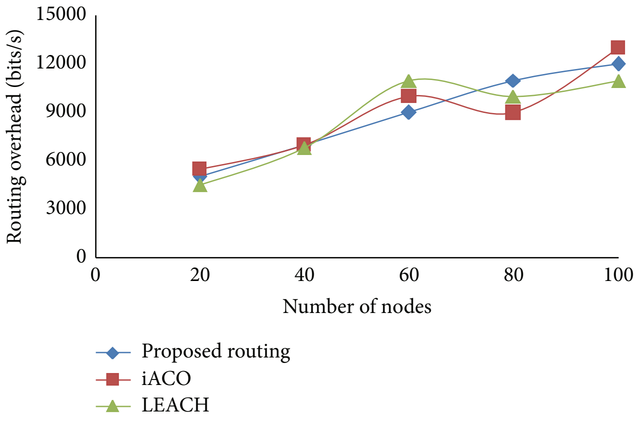


Figure 8: Routing overhead comparison.

4. Performance Evaluation

The performance of ant-based secure routing protocol for WSN is simulated using realistic scenarios. We simulated a WSN having 100 nodes using NS-2. These nodes are randomly deployed in the area of 100 by 100 meters. Initial node energy is set to 6 mJ. Maximum distance between nodes is not more than 20 meters. Each data packet is of 200 bytes. We compared our routing mechanism with low energy adaptive clustering hierarchy (LEACH) and iACO [41]. Figure 5 shows the network lifetime comparison of three routing protocols based on number of rounds.

Initially, the performance of iACO is better than the other; however, as long as the number of nodes increases, iACO shows performance degradation. The proposed routing protocol shows consistent performance in all types of nodes densities. However, overall iACO and LEACH performance is better as compared to our proposal. The reason is that, iACO and LEACH, both are designed to efficiently utilize available energy, while the design goal of our protocol is to ensure security.

In Figure 6, route discovery efficiency in a network of 100 nodes is presented. In this experiment, the performance of iACO is better than the other candidate solutions. The reason is that iACO finds shortest optimal path. LEACH protocol takes more time to find optimal path from source to destination. The reason is that, in LEACH, first data is forwarded to cluster head and then cluster head forwards data to destination after performing data aggregation.

In Figure 7, end to end delay is presented which shows some interesting results. The proposed routing scheme has less end to end delay as compared to LEACH and iACO. The reason is that the proposed routing scheme uses two paths for data transmission, while the other two routing protocols use single path.

Figure 8 compares routing overheads of all three routing protocols with 100 nodes.

All routing protocols have almost similar routing overheads. The proposed routing protocol creates routing overheads due to computation of reputation values, while the rest of two routing protocols create routing overheads by considering energy consumption.

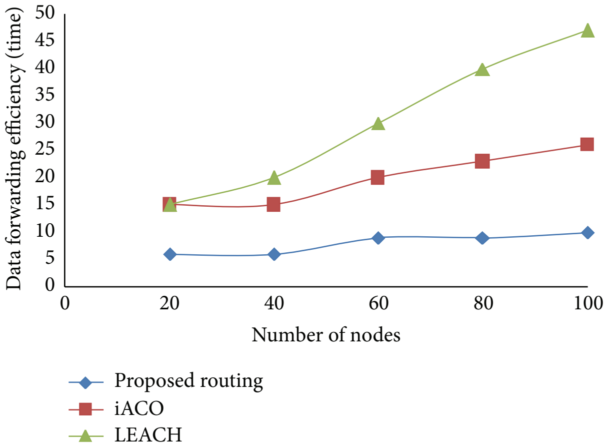


Figure 9: Data forwarding efficiency.

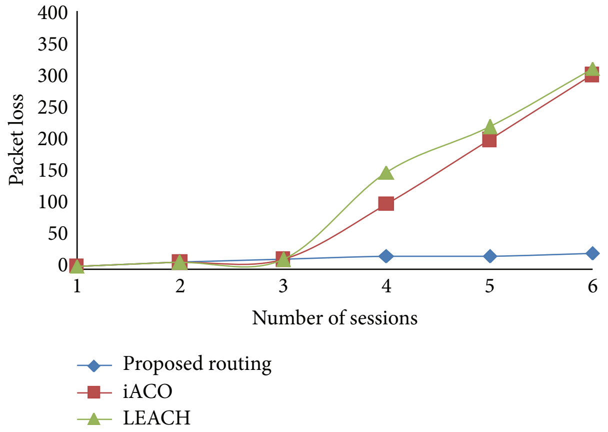


Figure 10: Packet loss in presence of malicious nodes.

In Figure 9, data forwarding efficiency of all protocols is presented. The proposed routing scheme is more efficient and delivers data from source to destination more quickly due to maintenance of two paths.

In Figure 10, few malicious nodes are introduced to observe the packet loss ratio of all the routing protocols. The malicious nodes are used to either drop or misdirect the traffic between source and destination. This simulation is set up in many sessions. Every session is used to forward 400 packets toward destination node. Malicious nodes are introduced from third session onward (i.e., in sessions 4, 5, and 6). In first three sessions, all the routing protocols successfully forward all the packets without any loss. However, when malicious nodes are introduced, most of the packets are either dropped or misdirected by the other routing schemes. However, our proposed mechanism successfully delivered most of the data.

Our routing mechanism is more efficient in presence of malicious nodes due to security mechanism. On the other hand, iACO and LEACH cannot distinguish malicious node in their path, and, that is why packet loss or packet misdirection ratio is very high.

5. Conclusion

Most of WSNs are deployed in harsh and hostile environment, so some sort of security mechanism is highly desirable. Secure routing protocol is an efficient way to ensure security in data forwarding from source to destination. In this paper, we presented in detail a secure routing protocol for WSN which is based on ant colonization technique. We use hello packets for surrounding neighbor's discovery. Our mechanism uses forward ants which collect and increment the reputation values along the path. Similarly, destination node uses backward ants which carry information and instruction from destination node about route security. The proposed mechanism uses two paths for data forwarding not only to overcome the problem of node failure but also to increase the efficiency of overall network.

When compared to other routing protocols such as iACO and LEACH, our proposed routing scheme shows better performance in terms of end to end delay, routing overheads, and data forwarding efficiency. Furthermore, the proposed mechanisms show high data delivery rate in the presence of malicious nodes.

Acknowledgments

The authors extend their appreciation to the Research Centre, College of Applied Medical Sciences, and the Deanship of Scientific Research at King Saud University for funding this research.

References

[1] S. Khan, N. Mast, K. Loo, and A. Silahuddin, "Cloned access point detection and prevention mechanism in IEEE 802.11 wireless mesh networks," *International Journal of Information Assurance and Security*, vol. 3, no. 4, pp. 257–262, 2008.

[2] N. Alrajeh, S. Khan, J. Lloret, and J. Loo, "Secure routing protocol using cross-layer design and energy harvesting in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 374796, 11 pages, 2013.

[3] N. Alrajeh, S. Khan, and B. Shams, "Intrusion detection systems in wireless sensor networks: a Review," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 167575, 7 pages, 2013.

[4] S. Khan and K.-K. Loo, "Real-time cross-layer design for large-scale flood detection and attack trace-back mechanism in IEEE 802.11 Wireless Mesh Networks," *Elsevier Network Security*, vol. 2009, no. 5, pp. 9–16, 2009.

[5] S. Khan, N. A. Alrajeh, and K.-K. Loo, "Secure route selection in wireless mesh networks," *Computer Networks*, vol. 56, no. 2, pp. 491–503, 2012.

[6] S. Khan, K. Loo, R. Comley, and A. N. Khwildi, "Surveying Ad hoc and Secure Routing in wireless mesh networks," *International Journal of Information Assurance and Security*, vol. 6, no. 1, pp. 73–80, 2011.

[7] S. Khan, K.-K. Loo, N. Mast, and T. Naeem, "SRPM: secure routing protocol for IEEE 802.11 infrastructure based wireless mesh networks," *Journal of Network and Systems Management*, vol. 18, no. 2, pp. 190–209, 2010.

[8] S. Khan, K. Loo, and Z. U. Din, "Cross layer design for routing and security in multi-hop wireless networks," *International Journal of Information Assurance and Security*, vol. 4, no. 2, pp. 170–173, 2009.

[9] S. Khan and K. Loo, "Cross layer secure and resource-aware on-demand routing protocol for hybridwireless mesh networks," *Wireless Personal Communications*, vol. 62, no. 1, pp. 201–214, 2012.

[10] N. Meghanathan, "A survey on the communication protocols and security in cognitive radio networks," *International Journal of Communication Networks and Information Security*, vol. 5, no. 1, pp. 19–38, 2013.

[11] M. Radi, B. Dezfouli, K. A. Bakar, and M. Lee, "Multipath routing in wireless sensor networks: survey and research challenges," *Sensors*, vol. 12, no. 1, pp. 650–685, 2012.

[12] K. Saleem, N. Fisal, S. Hafizah, S. Kamilah, and R. A. Rashid, "Ant based self-organized routing protocol for wireless sensor networks," *International Journal of Communication Networks and Information Security*, vol. 1, no. 2, pp. 42–46, 2009.

[13] M. Frederickson, *A Publication of the National Electronics Manufacturing Center of Excellence*, 2005.

[14] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[15] S. Singh, M. Singh, and D. Singh, "Routing protocols in wireless sensor networks, a survey," *International Journal of Computer Science & Engineering Survey*, vol. 1, pp. 25–34, 2010.

[16] A. M. Popescu, G. I. Tudorache, B. Peng, and A. H. Kemp, "Surveying position based routing protocols for wireless sensor and Ad-hoc networks," *International Journal of Communication Networks and Information Security*, vol. 4, no. 1, pp. 41–67, 2012.

[17] O. Fdili, Y. Fakhri, and D. Aboutajdine, "Impact of queue buffer size awareness on single and multi service real-time routing protocols for WSNs," *International Journal of Communication Networks and Information Security*, vol. 4, pp. 104–111, 2012.

[18] M. Hussaini, H. Bello-Salau, A. Salami, F. Anwar, A. Abdalla, and M. Islam, "Enhanced clustering routing protocol for power-efficient gathering in wireless sensor network," *International Journal of Communication Networks and Information Security*, vol. 4, pp. 18–28, 2012.

[19] A. Kellner, O. Alfandi, and D. Hogrefe, "A survey on measures for secure routing in wireless sensor networks," *International Journal of Sensor Networks and Data Communications*, vol. 1, pp. 1–17, 2012.

[20] J. Sen, "A survey on wireless sensor network security," *International Journal of Communication Networks and Information Security*, vol. 1, pp. 55–78, 2009.

[21] K. Xing, "Attacks and countermeasures in sensor networks, a survey," *Springer Network Security*, vol. 7, pp. 534–548, 2005.

[22] V. Kesavan and S. Radhakrishnan, "Multiple secret keys based security for wireless sensor networks," *International Journal of Communication Networks and Information Security*, vol. 4, pp. 68–76, 2012.

[23] X. Wei, J. Fan, M. Chen, T. Ahmed, and A. K. Pathan, "SMART: a subspace based malicious peers setection algorithm for P2P systems," *International Journal of Communication Networks and Information Security*, vol. 5, no. 1, pp. 1–9, 2013.

[24] M. Azeem, K. Khan, and A. Pramod, "Security architecture framework and secure routing protocols in wireless sensor networks-survey," *International Journal of Computer Science & Engineering Survey*, vol. 2, pp. 189–204, 2011.

[25] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[26] B. Kur, *Secure routing protocols for wireless sensor networks [M.S. thesis]*, University Faculty of Informatics, 2008.

[27] P. Samundiswary, D. Sathian, and P. Dananjayan, "Secured greedy perimeter stateless routing for wireless sensor networks," *International Journal of Ad Hoc, Sensor & Ubiquitous Computing*, vol. 1, pp. 9–20, 2010.

[28] G. Kumar, I. Titusb, and S. I. Thekkekarab, "A comprehensive overview on application of trust and reputation in wireless sensor network," *Procedia Engineering*, vol. 38, pp. 2903–2912, 2012.

[29] D. Khurana and M. Singla, "Secure and authenticated source routing in wireless networks," *International Journal Of Computer science*, vol. 12, no. 3, 2012.

[30] Z. Che-Aron, W. F. M. Al-Khateeb, and F. Anwar, "ENFAT-AODV: the fault-tolerant routing protocol for high failure rate wireless sensor networks," in *Proceedings of the 2nd International Conference on Future Computer and Communication (ICFCC '10)*, pp. V1467–V1471, May 2010.

[31] S. Khan, K.-K. Loo, and Z. U. Din, "Framework for intrusion detection in IEEE 802.11 Wireless Mesh Networks," *International Arab Journal of Information Technology*, vol. 7, no. 4, pp. 435–440, 2010.

[32] S.-B. Lee and Y.-H. Choi, "A secure alternate path routing in sensor networks," *Computer Communications*, vol. 30, no. 1, pp. 153–165, 2006.

[33] V. Thiruppathy Kesavan and S. Radhakrishnan, "Multiple secret keys based security for wireless sensor networks," *International Journal of Communication Networks and Information Security*, vol. 4, no. 1, pp. 68–76, 2012.

[34] M. S. Lin, J. S. Leu, W. C. Yu, and K. H. Li, "TBRA: termite based routing algorithm in 3D wireless sensor networks," in *Proceedings of the IEEE 75th Vehicular Technology Conference*, pp. 1–5, 2012.

[35] P. G. Hoolimath, M. Kiran, and G. R. Mohana Reddy, "Optimized tERMITE: a bio-inspired routing algorithm for MANET's," in *International Conference on Signal Processing and Communications (SPCOM '12)*, 2012.

[36] J. A. P. Martins, S. L. O. B. Correia, and J. C. Júnior, "Ant-DYMO: a bio-inspired algorithm for MANETS," in *Proceedings of the 17th International Conference on Telecommunications (ICT '10)*, pp. 748–754, April 2010.

[37] G. A. Di Caro, F. Ducatelle, and L. M. Gambardella, "AntHocNet: an ant-based hybrid routing algorithm for mobile ad hoc networks," in *Proceedings of the Parallel Problem Solving from Nature (PPSN '04)*, vol. 3242 of *Lecture Notes in Computer Science*, pp. 461–470, Springer, 2004.

[38] L. J. G. Villalba, D. R. Cañas, and A. L. S. Orozco, "Bio-inspired routing protocol for mobile ad hoc networks," *IET Communications*, vol. 4, no. 18, pp. 2187–2195, 2010.

[39] A. Cristina, B. Kochem Vendramin, A. Munaretto, M. Regattieri Delgado, and A. Carneiro Viana, "A Greedy Ant Colony Optimization for routing in delay tolerant networks," in *GLOBECOM Workshops Computing and Processing*, pp. 1127–1132, December 2011.

[40] S. S. Iyengar, H.-C. Wu, N. Balakrishnan, and S.Y. Chang, "biologically inspired cooperative routing for wireless mobile sensor networks," *IEEE System Journal*, vol. 1, no. 1, pp. 29–37, 2007.

[41] V. Mahadevan and F. Chiang, "iACO: a bio inspired power efficient routing scheme for sensor networks," *International Journal of Computer Theory and Engineering*, no. 6, pp. 1793–8201, 2010.

[42] L. B. Ribeiro and M. F. De Castro, "BiO4SeL: a bio-inspired routing algorithm for sensor network lifetime optimization," in *Proceedings of the 17th International Conference on Telecommunications (ICT '10)*, pp. 728–734, April 2010.

[43] N. Chauhan, A. Nain, and D. Srivastava, "A bio-inspired energy efficient routing approach to resolve broken link problem in WSN," *International Journal of Computer Applications*, vol. 48, no. 25, pp. 18–24, 2012.

[44] S. S. Iyengar, H. C. Wu, N. Balakrishnan, and S. Y. Change, "Biologically inspired cooperative routing for wireless mobile sensor networks," *IEEE Systems Journal*, no. 1, pp. 29–37, 2007.

[45] S. Ganeriwal and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," in *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, pp. 66–77, October 2004.

[46] K. K. Lavania, S. M. Tiwari, and S. Batra, "Data encryption in the hostile environment for wireless sensor network using virtual energy and trigger time response protocol," *International Journal of Computer Science Issues*, vol. 8, no. 3, pp. 538–542, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 521497, 20 pages
http://dx.doi.org/10.1155/2013/521497

Research Article

On Lightweight Intrusion Detection: Modeling and Detecting Intrusions Dedicated to OLSR Protocol

Mouhannad Alattar,[1] Françoise Sailhan,[2] and Julien Bourgeois[1]

[1] *UFC/FEMTO-ST Institute, UMR CNRS 6174, 25201 Montbéliard, France*
[2] *Cédric Laboratory, CNAM, 75003 Paris, France*

Correspondence should be addressed to Mouhannad Alattar; mouhannad.alattar@univ-fcomte.fr

Received 1 March 2013; Accepted 23 April 2013

Academic Editor: S. Khan

Mobile ad hoc networks mostly operate over open, adverse, or even hostile environments and are, therefore, vulnerable to a large body of threats. Conventional ways of securing network relying on, for example, firewall and encryption, should henceforth be coupled with advanced intrusion detection. To meet this requirement, we first identify the attacks that threaten ad hoc networks, focusing on the Optimized Link State Routing Protocol. We then introduce IDAR, a signature-based Intrusion Detector dedicated to ad hoc routing protocols. Contrary to existing systems that monitor the packets going through the host, our system analyses the logs so as to identify patterns of misuse. This detector scopes with the resource-constraints of ad hoc devices by providing distributed detection; in particular, depending on the level of suspicion and gravity, in-depth cooperative diagnostic may be launched. Simulation-based evaluation shows limited resource consumption (e.g., memory and bandwidth) and high detection rate along with reduced false positives.

1. Introduction

Securing mobile [Ad hoc] networks (MANETs for short) is particularly challenging because these networks often operate in adverse or even hostile environments [1, 2]. In addition, they are characterized by the open radio-based medium of communication [3], the dynamic topology [4, 5], the lack of centralized administration/security enforcement points (e.g., switches and routers) [6], the low degree of physical security of the mobile nodes, and the limited resources (e.g., energy, bandwidth) [7]. Hence, MANET is much more vulnerable to attacks than the traditional infrastructure-based network [8]. Conventional prevention techniques, for example, firewall, encryption [9], and authentication [10], cannot totally eliminate the attacks. Indeed, they address the outsiders but they are less helpful for protecting against the insiders; if a legitimate node becomes compromised, then all the secrets associated with this latter would be open to attackers. Moreover, new attacks emerge and find usually a way to penetrate the aforementioned techniques. Therefore, there is a need for a reactive mechanism, such as Intrusion Detection Systems (IDSs for short) [11], which constitutes a second line of defense.

As a first step upon this goal, we survey the attacks that have been reported in the literature. More specifically, we focus on the attacks targeting the Optimized Link State Routing (OLSR for short) [12]; its central role, namely, determining multi hop paths among devices, designates the ad hoc routing protocol as one of the favorite targets for the attackers [13]. Our attempt is not restricted to describing a bunch of attacks. Instead, we categorize and detail each attack relying on a representation/formalism that captures the complexity and temporal dependencies between each of the constituting subtasks. While describing attack, we attempt to circumvent the general form of this attack so as to keep to a minimum the detection that fails due to a varying attack. Based on these modeled attacks, we further implement one attack, challenge and derive appropriate intrusion detection.

Recent works show that attack may be identified by detecting a deviation to the correct behavior (anomaly detection); this correct behavior is either hand-specified relying on a protocol description (typically a Request For

Comment (RFC for short)) as in, for example, [14] or automatically built/analyzed using machine learning or data mining techniques, for example, [15]. The difficulty involved in automatically modeling the behavior of such dynamic network leads to a large number of false positives that may be reduced by coupling automatic and specification-based anomaly detection [16]. Moreover, the used techniques are mostly characterized by their intensive calculation and, henceforth, the extensive consumption of resources [2]. An alternative consists in describing the way an intruder penetrates the system (by establishing intrusion signature) so as to detect any behavior that is close to this previously defined signature. Little attention has been centered on signature-enabled detection (also called misuse detection) in MANET [17], although it guarantees a high rate of detection along with a limited number of false alarms [18].

We propose IDAR, a log-, signature-based Intrusion Detector dedicated to ad hoc routing protocols. IDAR does not necessitate sniffing/inspecting the traffic as it is the case with the majority of other IDSs. Thus, it avoids the permanent strain of energy, bandwidth [19], and computational power [20] that accompanies traffic sniffing and analyzing. Our system rather takes advantage of the audit logs that are generated by the routing protocol so as to detect evidences of intrusion attempts. In practice, a sequence of events are extracted from logs so as to be matched against a set of predefined intrusion signatures—a signature is thought of as a pattern of events that characterizes an intrusion.

Main challenges stem from the need to keep to a minimum the number of diagnostics and the computational load related to the intrusion identification while minimizing the traffic generated when gleaning intrusion evidences. This calls for developing a lightweight and distributed intrusion detection system that scopes with the cooperative nature of ad hoc networks and the device resource constraints. Towards this goal, IDAR is designed to be a distributed and cooperative detection system, which parses logs as close as possible from the device that generates them so as to diminish the number of long-distant communications. Furthermore, we propose to categorize the intrusion evidences according to their level of gravity/suspicion. Such categorization enables us to carefully plane the diagnostic; an in-depth diagnostic is initiated only when a sufficient degree of suspicion exists and terminated as soon as a result is obtained. Thus, we guarantee keeping to a minimum the number/the duration of diagnostics. The performance of IDAR is evaluated in a simulated MANET that is coupled with virtual machines. Such coupling permits us measuring both the detection accuracy and the amount of consumed resources.

The reminder of this paper is organized as follows. We first survey attacks on ad hoc network, breaking down the successive steps that characterize conquering attacks (Section 2). Grounded upon the defined intrusion signatures, we present a distributed, log- and signature-based intrusion detection system (Section 3) and evaluate its performance (Section 4). Then, we conclude this paper with a summary of our results along with directions for future works (Section 6).

2. Vulnerabilities

Physical, data link, and network layers are all subject to vulnerabilities. Whereas in the physical and data link layers, vulnerabilities are common to IEEE 802.11 wireless networks, vulnerabilities in network layer are specific to ad hoc networks. More specifically, Zeroconf and routing protocols constitute the main target of attacks [13]. The reason is threefold. First, no security countermeasure is specified or implemented as a part of the drafts or RFCs (request for comments) proposed through the Internet Engineering Task Force MANET (http://www.ietf.org/dyn/wg/charter/manet-charter.html) or Zeroconf (http://www.zeroconf.org/) working groups. Second, the absence of a centralized infrastructure complicates the deployment of preventive measures, for example, firewalls or key/authentication infrastructures. Third, any device may operate as router, which facilitates the manipulation of multihops messages as well as the compromising of the routing functionality. Attacks on those protocols fall into two main categories, passive versus active [21]. With the former, an intruder intercepts the traffic in order to reveal useful information (e.g., the roles played by the nodes and their locations) whereas with the latter, an unauthorized action is attempted [22]. Active attacks are further sub-classified according to their unauthorized actions as follows [23].

(i) *Drop attack* consists in dropping routing message(s).

(ii) *Modify and forward attack* modifies received routing message(s) before forwarding it.

(iii) *Active forge attack* proactively generates deceptive routing message(s).

Although the above attacks threaten both routing and Zeroconf protocols, we hereafter concentrate on routing protocol and illustrate our presentation by exemplifying attacks on a proactive protocol called OLSR.

2.1. Background on the OLSR Protocol. OLSR aims to maintain a constantly updated view of the network topology on each device. One fundamental is the notion of multipoint relay (MPR): each device selects a subset of the 1-hop neighbors, called MPRs, that is responsible for forwarding the control traffic in the entire network. The basic idea is to select the minimum number of MPRs that cover the two-hops neighbors so as to reduce the number of nodes retransmitting control messages and hence keep to a minimum the bandwidth overload. Note that a redundant MPR may be selected as to increase the reachability but this leads to increase the overhead. In practice, a node N selects the MPRs among the 1-hop neighbors that are announced (in addition, link layer information provided by, e.g., the IEEE 802.11 protocol, may be used by a node to update its own routing tables) in periodic heartbeat messages, termed hello messages. Then, a *Topology Control* message (TC for short), intended to be diffused in the entire network (the message Time To Live field (TTL for short) can be used so as to limit message diffusion), is created by the selected MPR(s). In TC message, a MPR declares the nodes (including N) that selected itself to act as MPR. Thanks to TC messages, any device computes the shortest path (in term of the number of hops) to any destination, such path being represented as a sequence of MPRs. In addition to the above, last versions of the protocol specification support a node holding several network interfaces which are declared (if many) in a so-called MID (Multiple Interface Declaration) message. This message is broadcasted on a regular basis by MPRs so that one another maps multiple interfaces of a given node with the main address of this latter, hence permitting a unique identification. The aforementioned functionalities compose the core of OLSR. Additional extensions have been devised in compliance with the above-summarized OLSR specification. Examples include (i) dealing with the nodes that commit (or not) to carry the traffic for others and (ii) supporting interconnection of an OLSR MANET with another routing domain, for example, OSPF-enabled routing domain. With the former, node advertises (in hello message) its willingness to carry/forward traffic. With the latter, OLSR is extended to import (and, resp., export) the routes provided by other routing protocols (resp., OLSR). For this purpose, any gateway with associated host(s) and/or network(s) generates periodically a HNA message including those host(s) and/or network(s) (i.e., the related network address and the netmask); this message is further disseminated by MPRs. Such an auxiliary function is enabled by providing a basic layout of any OLSR packet and by ignoring unknown (and hence not handled) packets. Overall, these core and auxiliary functionalities are together subject to a variety of attacks.

Recently, a new version of OLSR (so-called OLSRv2) is specified in an Internet-draft [24]. OLSRv2 distinguishes itself from its predecessor by considering the link metric rather than hop count for selecting the shortest path. It is supposed that OLSRv2 would have more modular and flexible architecture that facilitates add-ons extensions for, for example, security, QoS, and multicast [25]. It will also possess a simplified packet format and reduced-size messages due to address compression [26]. However, both, OLSR and OLSRv2, retain the same basic algorithms and mechanisms.

It is worth mentioning that several enhanced versions of OLSR have been proposed. Some of them tackle the security issue and use signature, hash chain, or encryption schemes in order to provide the security to OLSR [27, 28]. In general, these versions aim to ensure (i) the integrity of the routing information, that is, preventing the unauthorized modifications of the routing messages by the intermediate nodes on the path from the source to the destination and (ii) the node-to-node authentication in order to prevent identity spoofing. However, they do not address, for example, the case where the source node is itself compromised and hence it generates incorrect routing messages. Thus, they are still incapable of preventing some types of attack [29]. Therefore, there is always a need to employ a detection mechanism so as to handle the attacks that would success penetrating the prevention techniques.

2.2. Attacks Targeting the OLSR Protocol.

2.2. Attacks Targeting the OLSR Protocol. The attacks threatening OLSR are hereafter detailed and classified according to the model introduced in [30]. This model provides the level of expressiveness necessary to specify the relationship between the actions and their related consequences. We further enrich this model with temporal annotations (Table 1). As a consequence, complex attacks, their constituting actions and consequences are temporally and successfully depicted and categorized as drop attack (Section 2.2.1), active forge attack (Section 2.2.2), and modify and forward attack (Section 2.2.3).

Table 1: Notations.

Communication	
$Y \xrightarrow{M_t} X$	At t, Y sends a message M that is received by X
$Y \xrightarrow{M_t}$	At t, Y sends a message M
$Y \overset{M_t}{\nrightarrow}$	Y does not send a message M at t
$\xrightarrow{M_t} X$	X receives a message M at t
$\overset{M_t}{\nrightarrow} X$	X does not receive a message M at t
Parameters	
$\triangle t, \triangledown t$	Period of time
$\mathscr{I}$	Set of malicious nodes
NS_X	Set of 1-hop neighbors of X
sq	Sequence number
hc	Hop count
w_X	Willingness of X to forward packets on behalf of others
MPR_x	MPRs of X
$\mathrm{Sel}_{\mathrm{MPR}_x}$	MPR selectors set of X
Messages	
Hello	Hello message
TC	Topology Control message
MID	Multiple Interface Declaration
HNA	Host and Network Association
CM	Control Message
RM	Received Control message
FM	Control message intended to be forwarded

2.2.1. Drop Attack. In practice, a drop attack consists of an intruder that drops a control message instead of relaying it. This dropping has an impact only if the dropped control message is intended to be forwarded; as illustration, a suppressing of a hello message that is broadcasted over one hop, has no consequence. Thus, with OLSR, threatened messages are restricted to the messages that are created and/or broadcasted by a MPR so as to be rediffused by other MPRs, that is, Topology Control (TC), Multiple Interface Declaration (MID), and Host and Network Association (HNA) messages. More particularly, let us consider a host S that sends a control message which is intended to be forwarded. This message, which is originated at t ($S \xrightarrow{\mathrm{FM}_t} I$), is received by I that drops it. In practice, I drops a control message if I does not forward this message during a period $|t' - t|$ less than the maximum allowed (any control message is characterized by an emission interval as well as a holding time in the routing tables. These two related values participate in setting a validity time that should be herein considered in conjunction with the message type) period $\triangle t$:

$$S \xrightarrow{\text{FM}_t} I, I \overset{\text{FM}_{t'}}{\nrightarrow}, \left|t' - t\right| > \triangle t \\ \Downarrow \\ I \in \mathscr{I}. \tag{1}$$

Such a dropping is naturally applied on any packet that is empty, expired, duplicated, or out of order. In addition, restrictive forwarding is provided, meaning that in practice (only) the 1-hop neighbor(s) of S that have been selected by S to act as MPR(s), forward S's messages. Apart from the aforementioned conditions that lead to a convenient dropping, the remaining reason motivating a dropping is threefold. An intruder (or a compromised node), a selfish or a malfunctioning node disturbs the routing calculation by dropping a control message that should be forwarded. The attempt to drop any packet is termed *black hole*. A selective dropping (named *gray hole*) is detected by taking into account additional fine-grained or discriminative criteria, including the choice of specific final/1-hop-away destination or source, percentage/rate of packets received or forwarded, and message type. Rather than dropping the traffic, an opposite misbehavior consists in introducing falsified routing information.

2.2.2. Active Forge Attack. An active forge attack comes from a node that introduces novel deceptive routing messages. Among others, the *broadcast storm* stems from forging control messages so as to exhaust resources (e.g., energy) and saturate the communication medium. For this purpose, an intruder I forges a large number of control messages CM within a short period of time $\triangledown t$ (see Expression 2). This attack may be local (e.g., targeting a 1-hop neighborhood) or global (e.g., relying on the multihops broadcasted messages). It may also be conducted in a distributed manner with several colluding nodes so as to simultaneously emit (a large number of) control messages. Note that in order to prevent a broadcast storm resulting from the involuntary synchronization of some emitting nodes, a delaying of the control messages is recommended in the RFC; a jitter (typically, a random value) is subtracted from the interval at which control messages are generated (and also forwarded):

$$I \xrightarrow{\text{CM}_t}, I \xrightarrow{\text{CM}'_t}, \left|t' - t\right| < \triangledown t \\ \Downarrow \\ I \in \mathscr{I}. \tag{2}$$

A special type of *broadcast storm* is *routing table overflow* attack that threatens especially the proactive routing protocols (e.g., OLSR) as they periodically update the routing information [31]. Herein, one or several colluding intruders prevent the well-behaving nodes from discovering new existing routes by dumping the network with route advertisements to nonexisting nodes. In general, *broadcast storm* constitutes a kind of denial of service that is characterized by a high visibility. Therefore, it is typically combined to masquerading—although less intrusive attacks may also be preferred. In practice, masquerading lies in sending a control message CM including a switched identification ($I \xrightarrow{\text{CM}(I)_t}, I \xrightarrow{\text{CM}(S)_t}$). Note that this case is distinguished from a node that holds several interfaces and that advertises those later in a dedicated MID message. Apart from masquerading a denial of service, identity spoofing may be intended to create conflicting route(s) and potentially routing loop(s). In this latter case, spoofing the identity of a node S, belonging to the network, is necessary. As pointed out in [32], the spoofing attack may also be coupled with a modification of the willingness field so as to impact the MPR selection (Expression 3). In practice, if I emits a hello message including an originator address already assigned to another node S ($I \xrightarrow{\text{hello}(S)} D$), then I modifies the local topology as seen by its 1- and 2-hops neighbors. In addition, if I modifies the willingness field w'_S of S (with $w'_S \neq w_S$), then the selection of S as MPR is impacted; recall that MPRs are selected among the nodes with highest willingness, and in case of multiple choices, the node which provides a reachability to the maximum number of nodes is primarily selected. For instance, a node whose willingness attribute set to $will\_never = 0$ (resp., $will\_always = 7$), is never (resp., always) selected as MPR; that is, $w'_S = will\_never \Rightarrow S \notin \text{MPR}_D$ (versus $w'_S = will\_always \Rightarrow S \in \text{MPR}_D$):

$$S \xrightarrow{\text{hello}(S,w_S)_t} D, I \xrightarrow{\text{hello}(S,w'_S)t'} D, \\ \left|t' - t\right| < \triangle t, w_S \neq w'_S \\ \Downarrow \\ I \in \mathscr{I}, \\ S \text{ is spoofed}, \\ w'_S = will\_never \Longrightarrow S \notin \text{MPR}_D, \\ w'_S = will\_always \Longrightarrow S \in \text{MPR}_D. \tag{3}$$

Active forge attacks are not restricted to identity spoofing (possibly coupled with a tampering of the willingness field). They also cover the tampering of control messages including incorrect adjacent links (hello messages), topology information (TC messages), and network interfaces (MID and HNA messages). In the following, we detail each of those. In the first case (Expression 4), I forges a hello message, which declares a list of 1-hop and symmetric neighbors NS'_I differing from the real set NS_I. A symmetric 1-hop neighbor, or simply symmetric neighbor, corresponds to an adjacent node with which communication is bidirectional. In practice, it means that the hello message exchanged by two nodes is heard by both of these later:

$$\begin{gathered} I \xrightarrow{\text{hello}\left(NS_I'\right)_t} S, NS_I' \neq NS_I \\ \Downarrow \\ I \in \mathcal{I}. \end{gathered} \tag{4}$$

Whenever forging the set of neighbors NS_I', the attacker has three options.

(1) Declaring a nonexisting node as a symmetric neighbor implies that I (or another misbehaving node) is further selected as MPR (Expression 5). Indeed, if I advertises a non-existing node N ($N \notin \mathcal{N}$ with $\mathcal{N}$ defining the set of nodes composing the OLSR network (according to the OLSR RFC [12], messages can be flooded into the entire network (with a maximum network diameter defined by the message Time To Live field, TTL for short), or flooding can be limited to nodes within a diameter (defined in terms of number of hops) from the originator of the message. For the sake of clarity, let $\mathcal{N}$ represent the network in both cases)), I ensures that no other (well-behaving) MPR claims being a 1-hop symmetric neighbor of N. Recall that the set of MPRs is selected so that all the 2-hops and symmetric neighbors are covered; I is hence selected as MPR:

$$\begin{gathered} S \xrightarrow{\text{hello}\left(NS_S\right)_t}, I \xrightarrow{\text{hello}\left(NS_I'\right)_{t'}} S, \left|t' - t\right| < \triangle t, \\ \exists N \in NS_I' \ni\colon N \notin \mathcal{N} \cap NS_I \\ \Downarrow \\ I \in \mathcal{I}, \\ \exists I' \in \mathcal{I} \cap NS_S \ni\colon I' \in \mathrm{MPR}_S, \\ \mathrm{Card}\left(NS_I' \setminus \left(NS_I' \cap \mathcal{N}\right)\right) > 0. \end{gathered} \tag{5}$$

This assertion is verified as long as no other misbehaving neighbor of S is claiming the same. Overall, inserting at least one non-existing neighbor ($\exists N \in NS_I' \ni\colon N \notin \mathcal{N} \cap NS_I$) guarantees that a misbehaving node I' (with $I' \in \mathcal{I}$) is selected to act as MPR of S ($\exists I' \in \mathcal{I} \cap NS_S \ni\colon I' \in \mathrm{MPR}_S$). In addition to the above, the connectivity of I is also artificially increased ($\mathrm{Card}(NS_I' \setminus (NS_I' \cap \mathcal{N})) > 0$).

(2) Declaring that an existing node is a symmetric 1-hop neighbor whereas that node is far away (i.e., is not a neighbor) or is a neighbor but that is not symmetric ($\exists X \in NS_I' \cap \mathcal{N} \ni\colon X \notin NS_I$). This claiming increases artificially the connectivity of I, that is, $\mathrm{Card}((NS_I' \setminus NS_I) \cap \mathcal{N}) > 0$:

$$\begin{gathered} S \xrightarrow{\text{hello}\left(NS_S\right)_t}, I \xrightarrow{\text{hello}\left(NS_I'\right)_{t'}} S, \left|t' - t\right| < \triangle t, \\ \exists X \in NS_I' \cap \mathcal{N} \ni\colon X \notin NS_I \\ \Downarrow \\ I \in \mathcal{I}, \\ \mathrm{Card}\left(\left(NS_I' \setminus NS_I\right) \cap \mathcal{N}\right) > 0, \\ \nexists A \in \mathcal{N} \setminus \mathcal{I} \ni\colon A \in NS_S \wedge X \in NS_A \\ \Downarrow \\ \exists I' \in \mathcal{I} \ni\colon I' \in \mathrm{MPR}_S. \end{gathered} \tag{6}$$

If no (well-behaving) MPR covers S ($\nexists A \in \mathcal{N} \setminus \mathcal{I} \ni\colon A \in NS_S \wedge X \in NS_A$), then at least one misbehaving node (e.g., I) is selected as MPR of S ($\exists I' \in \mathcal{I} \ni\colon I' \in \mathrm{MPR}_S$). Such insertion typically characterizes an attempt to create a black hole; I introduces a novel path toward M that whenever selected provisions the black hole.

(3) Omitting an existing 1-hop neighbor and symmetric node, P ($\exists P \in NS_I \ni\colon P \notin NS_I'$), decreases artificially the connectivity of both P and I ($NS_I \nsubseteq NS_I'$). Note that if P has no other connectivity than the one obtained through a misbehaving node, P gets isolated:

$$\begin{gathered} S \xrightarrow{\text{hello}\left(NS_S\right)_t}, I \xrightarrow{\text{hello}\left(NS_I'\right)_{t'}} S, \left|t' - t\right| < \triangle t, \\ \exists P \in NS_I \ni\colon P \notin NS_I' \\ \Downarrow \\ I \in \mathcal{I}, \\ \exists I' \in \mathcal{I} \cap NS_S, NS_I \nsubseteq NS_I'. \end{gathered} \tag{7}$$

Overall, falsifying the neighboring adjacency by inserting (existing or nonexisting) neighbor(s) and/or omitting real neighbors potentially perverts the local topology seen by S (and more generally by one another) and impacts the selected MPR(s) by S. Nevertheless, note that in order to be selected as MPR of S (or to prevent its selection), there is no need for I to falsify the neighboring adjacency. Recall that, in a hello message, a field, termed *willingness*, designates the node's willingness to carry traffic on behalf of others; I hence prevents (resp., ensures) its selection as MPR, by simply setting its willingness field to the value *will_never* (resp., *will_always*). On the whole, the MPR selection is impacted by either falsifying the topological information included in hello message or by making use of the willingness attribute. Another (perhaps more straightforward) alternative refers to a slander node I declaring (resp., not declaring) itself as MPR although it has not (resp., has) been selected as MPR

[33]. For this purpose (Expression 8), I forges a *Topology Control* (TC) message including an incorrect set of 1-hop symmetric neighbors that have selected I as MPR (so-called MPR selector set). Depending on the level of redundancy required, a MPR advertises:

(1) the MPR selector(s) $\mathrm{Sel}_{\mathrm{MPR}_I}$ of I;

(2) the MPR selector(s) along with the MPR of I: $\mathrm{Sel}_{\mathrm{MPR}_I} \cup \mathrm{MPR}_I$;

(3) the 1-hop neighbors NS_I of I (hence including the MPR selectors and the MPR of I).

Let $\mathscr{A}_I$ represent the set advertised in the TC message: $\mathscr{A}_I = \mathrm{Sel}_{\mathrm{MPR}_I} \vee (\mathrm{Sel}_{\mathrm{MPR}_I} \cup \mathrm{MPR}_I) \vee NS_I$. This attack consists in I advertising an incorrect set $\mathscr{A}'_I$ differing from the real one $\mathscr{A}_I$:

$$I \xrightarrow{\mathrm{TC}(\mathscr{A}'_I)_t} S, \mathscr{A}'_I \neq \mathscr{A}_I$$
$$\Downarrow \tag{8}$$
$$I \in \mathscr{I}.$$

In particular, possible falsifications lie in either I inserting a non-existing node, or inserting an existing node but non-MPR selector, or omitting a node belonging to $\mathscr{A}_I$. Upon the reception of a falsified TC message, routing table is corrupted. This corruption contaminates the OLSR network and also any interconnected routing domain. Indeed, a node, well-behaving or not, acting as a gateway exports the wrong OLSR routes. Symmetrically, an intruder may also import incorrect routes to the OLSR domain. This latter attack, termed *sinkhole*, involves an intruder I defining itself as a gateway that provides an access to associated host(s) and/or network(s). This gateway generates periodically a HNA message including those host(s) and/or network(s) (i.e., the related address(es) and netmask(s)). This attack constitutes a generalization of the previously defined forging of corrupted TC messages; a node advertises either non-existing or existing but unreachable nodes or omits advertising reachable nodes. The similarity with the TC active forging lead us not detailing it hereafter.

The aforementioned forge attacks (e.g., link or route spoofing attacks, sinkhole) necessitate to tamper specific message fields while keeping this message syntactically correct. More generally, bogus control messages can be forged, hence creating an implementation-dependent effect. Generally speaking, similar tampering may be performed by an intruder that has acted as MPR prior forwarding a falsified control message.

2.2.3. Modify and Forward Attacks. Modify and forward attacks are characterized by an intermediate, which captures the victim's control message and replays, modifies, or drops this message before forwarding it. In the following, we depict the two former cases, ignoring the message dropping that has been already described. Replaying a control message includes delaying the emission of this message by recording and forwarding it later (potentially in another area) or repeating this message. As a consequence, routing tables are updated based on obsolete information. Note that each message contains a field indicating the period of time during which this message is considered as valid, and hence it is used to update the routing table. Both attacks can be systematic (i.e., targeting any multihops control traffic) or selective. They may also be performed in a distributed manner (Expression 9) with two intruders: one recording the control message from one region so as to replay it in another region (i.e., the one of the colluding intruder). Without loss of generality, let I_1 (resp., I_2) be the node that records it (resp., replays it):

$$S \xrightarrow{\mathrm{CM}(S)_t} I_1, I_1 \xrightarrow[\mathrm{enc}]{\mathrm{CM}(S)} I_2, I_2 \xrightarrow{\mathrm{CM}(S)_{t'}} Y,$$
$$S \notin NS_Y, \triangledown t \leq \left| t' - t \right| < \triangle t$$
$$\Downarrow \tag{9}$$
$$S \in NS_Y$$
$$\Downarrow$$
$$I_1, I_2 \in \mathscr{I}.$$

In practice, I_1 tunnels the control traffic by, for example, encrypting the routing message and/or relying on an alternative network interface (that may be advertised or not). Then, I_2 replays it. This attack leads to the creation of a *wormhole* whose length depends on the distance separating the two intruders. Note that, in order to stay invisible, both I_1 and I_2 may keep the identification field unchanged; the source is S rather than I_1 or I_2. This possibility corresponds to a masquerading. Sequence numbers constitute a standard mechanism that provides protection against replay attacks given that the sequence delivery is not required by OLSR. In counterpart, their usage can be hijacked so that the destination drops the message rather than using it. In practice, an intruder I increases (a wraparound mechanism is implemented; when the sequence number reaches an extreme boundary, it is reset to zero) (resp., decreases) the value of the sequence number sq ($S \xrightarrow{\mathrm{CM}(S)_{t,\mathrm{sq}}} I, I \xrightarrow{\mathrm{CM}(S)_{t',\mathrm{sq}'}},$ $\mathrm{sq}' \neq \mathrm{sq}$) so that the destination assumes that S is providing the freshest (resp., an obsolete) route and, therefore, ignores the subsequent (resp., the actual) control message(s):

$$S \xrightarrow{\mathrm{CM}(S)_{t,\mathrm{sq}}} I, I \xrightarrow{\mathrm{CM}(S)_{t',\mathrm{sq}'}}, \mathrm{sq}' \neq \mathrm{sq}, \left| t' - t \right| < \triangle t$$
$$\Downarrow \tag{10}$$
$$I \in \mathscr{I}.$$

An intruder I may also corrupt the routing table by maliciously modifying a received control message before forwarding it. This modification consists in tampering either the contents of the message or the identification of its source. The former case is similar to the forge attack, described above, wherein the intruder tampers the MPR selector set in TC message, the OLSR routes in HNA message or the interface(s) identification in MID message. While in the latter case, an intruder I forwards the packet containing the control message without changing the source address [34]. Consequently, two nodes consider themselves as 1-hop neighbors whereas they are not in the communication range of each other. An intruder may also disrespect the flooding mechanism in OLSR. This happens when the intruder retransmits a control message that is received from a non-MPR selector node [32]:

$$\begin{gathered} I \notin \mathrm{MPR}_S, S \xrightarrow{\mathrm{CM}_t} I, I \xrightarrow{\mathrm{CM}_{t'}}, \left|t' - t\right| < \triangle t \\ \Downarrow \\ I \in \mathcal{I}. \end{gathered} \tag{11}$$

Overall, attacks targeting OLSR protocol are classified into 3 types:

(i) *drop* attack that consists in totally or selectively dropping the control messages, for example, dropping TC message in order to foil route calculation;

(ii) *active forge* attack that attempts to poison OLSR's functionalities by introducing novel deceptive control message, for example, a hello message including an incorrect neighbor set so as to foil MPR selection;

(iii) *modify and forward* attack lies in an intruder that captures and modifies a control message before forwarding it (e.g., increasing the sequence number of a TC message; hence the subsequent TC messages are ignored).

An intruder launches the attack either in a standalone manner or in collusion with other intruder(s), constituting what so-called a *Byzantine* attack (e.g., *wormhole*), together usually coupled with masquerading. Detecting these attacks is far to be a trivial task because a minor deviation on an attack makes it undetectable. In addition, an attack can be composed of several subattacks. In order to tackle these issues, we describe/model the attack as general as possible, hence circumventing possible deviations. We then propose an intrusion detection system that detects composed attacks as much as their parties.

3. Intrusion Detection

We propose a distributed, log- and signature-based intrusion detection system named IDAR. In addition to providing a high detection accuracy, IDAR aims to maintain the available resources in MANET. For this purpose, the evidences of attack are extracted from the log and further classified according to their level of gravity. Such classification helps in planning the diagnostic and henceforth minimizing the computation overhead related to attack identification. During the diagnostic, evidences are matched to predefined intrusion signatures. The establishment of an intrusion signature and further the diagnostic operation are exemplified with the link spoofing attack we purposely developed.

3.1. Resource-Aware Evidence Gathering. IDAR distinguishes itself from other IDSs by extracting the signs of attack from the logs instead of sniffing the traffic. Thus, the consumption of energy and computational power resulting from evidence gathering and analyzing is minimized. In fact, sniffing traffic necessitates that the wireless network interface stays in mode promiscuous at all time, thus, the node can overhear all the packets within its transmission range. But promiscuous mode leads to consuming more energy and hence reducing rapidly the lifetime of the node/network. Indeed, a nondestination node in the radio range of either the sender or the receiver overhears some or all of their traffic. For the IEEE 802.11 MAC protocol, a nondestination node in discarding (i.e., nonpromiscuous) mode can enter into a reduced energy consumption mode and discard others' traffic [35]. Such mode requires less energy than the idle mode, which is the default mode in ad hoc network. While a nondestination node operating in promiscuous mode listens to all the traffic, whether or not it is the intended destination. The traffic is further received as if it was a broadcast traffic; thus, additional energy consumption is associated with promiscuous mode operation. Figure 1 represents a part of the experimental measurements realized in [19] about the energy consumption of an IEEE 802.11 2 Mbps wireless card. It shows the significant difference in energy consumption between promiscuous and nonpromiscuous modes. Note that the consumption in the nonpromiscuous mode is negative because it requires less energy than the reference idle mode. Besides increasing the amount of consumed energy, sniffing traffic imposes a huge computational overhead [20]. Indeed, the packet-level analysis, which is applied on the sniffed packets, strains significantly the available resources, that is, memory and CPU processing. Moreover, since all the traffic in the radio range is sniffed, many of the analyzed packets would be redundant and add nothing to the detection. In order to avoid this permanent strain of resources, IDAR does not sniff the traffic but it rather collects periodically the local logs. In particular, it focuses on the portion of logs that characterizes the

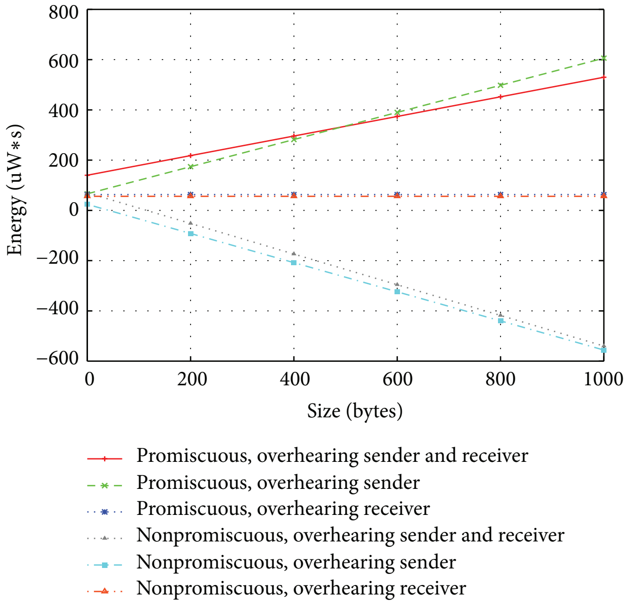


Figure 1: Energy consumption in promiscuous and nonpromiscuous modes.

activities of the routing protocol (e.g., packet reception, MPR selection). Note that additional logs, for example, system-, security-related logs, could be integrated and correlated. Once parsed, a log is used so as to detect a sign of suspicious activity. This consists in matching the log against a predefined intrusion signature. An intrusion signature is thought as a partially ordered sequence of events that characterizes an intrusion. Such procedure is potentially not only memory but also bandwidth consumer. Indeed, it involves examining local logs as well as performing an in-depth diagnostic where other nodes are requested so as to collect additional attack evidences, correlate, and match them against the defined intrusion signatures. The in-depth diagnostic offers a global view about the suspicious nodes, and therefore it increases the accuracy of the detection. But, it is a costly operation in terms of resources. Thus, the in-depth diagnostic must be carefully planned, that is, should be initiated only when a sufficient degree of suspicion exists and terminated as soon as a result is obtained. For this purpose, we propose to classify attack evidences so that depending on their level of gravity, the in-depth diagnostic may be performed. According to our classification, an evidence falls into one of the following four groups.

(i) *Initial-evidence group* contains the evidences that lead to launch an in-depth diagnostic over the network.

(ii) *Suspicious-evidence group* contains the evidences that lead to identify a node as suspicious.

(iii) *Confirmed-evidence group* contains the evidences that confirm the occurrence of an attack. This results in terminating the diagnostic and declaring the suspicious node as intruder.

(iv) *Cancel-evidence group* contains the evidences that eliminate the suspicion and stop the diagnostic.

These groups are populated with the evidences that are extracted from the log. If an evidence belonging to the *initial-evidence group* is discovered, then an in-depth diagnostic is launched so as to confirm (i.e., discovering an evidence belonging to *confirmed-evidence group*) or infirm (i.e., discovering an evidence belonging to the *cancel-evidence group*) the intrusion; both lead to the termination of the diagnostic. Relying on these groups, the evolution of any attack and its related detection is easily followed. In addition, its compact form facilitates the lightweight discovering of long-term intrusions.

3.2. Link Spoofing Attack. A link spoofing attack lies in falsifying hello message(s) so as to modify the local topology perceived by adjacent nodes. This attack influences the MPR selection. In particular, this attack owns a global impact: the MPR position provides to the intruder the possibility to eavesdrop, tamper, misrelay, or drop the traffic. As discussed in Section 2 (and further detailed in Expression 4), an intruder realizes a link spoofing attack through one of the following three cases.

(1) It advertises a non-existing and symmetric node. Thus, the intruder guarantees (unless another attacker advertises the same non-existing node) being selected as a MPR because this non-existing node is uniquely covered by the intruder.

(2) It advertises existing but nonneighboring node(s). The intruder is selected as a MPR if the advertised node(s) is (are) not already covered by another (well-behaving or malicious) MPR.

(3) It keeps under wraps neighboring and symmetric node(s). In this case, the connectivity of the intruder and consequently its chance of being selected as a MPR are both decreased. Used in a standalone manner, this attack aims to decrease the connectivity of one or several nodes; a complete isolation necessitates that no other (well-behaving) MPR covers that node(s).

We develop an attack (Expression 12) wherein an intruder falsifies a hello message that contains both a non-existing node (Case 1) and existing but non-neighboring nodes (Case 2). The reason that motivates this choice is twofold. First, by advertising a nonexisting node N, I ensures being selected as a MPR by the victim S ($\exists N \in NS'_I \ni: N \notin \mathcal{N} \cap NS_I$). Note that with the last versions of the RFC, a node (i.e., potentially an intruder) may dictate its selection as a MPR by advertising (in a hello message) its high willingness to relay messages. Nevertheless, previous versions (and their related implementations) ignore this case. Regardless of the version of the protocol, our intruder (as aforementioned previously) guarantees being selected as MPR by advertising a nonexisting and symmetric node. Second, by announcing common neighbors with another node L (Case 2), I increases the probability that L is not selected as a MPR (I replacing L) and henceforth increases its proper ascendancy. Note that the attack we developed does not consider the third case that involves the reduction of the intruder's connectivity. Nevertheless, attack signature deals with it:

$$\begin{gathered}
S \xrightarrow{\text{hello}\left(NS_S\right)_t} I, I \xrightarrow{\text{hello}\left(NS'_I\right)_{t'}} S, \left|t' - t\right| < \triangle t, \\
\exists N \in NS'_I \ni: N \notin \mathcal{N} \cap NS_I, \\
\exists L \in \text{MPR}_S \ni: \left[NS_L \setminus NS_I\right] \subseteq \left[NS'_I \setminus NS_I\right] \\
\Downarrow \\
I \in \mathcal{I}, L \notin \text{MPR}_S, \\
\exists I' \in \mathcal{I} \ni: I' \in \text{MPR}_S, \\
\text{Card}\left(NS'_I \setminus \left(NS'_I \cap \mathcal{N}\right)\right) > 0 \\
\text{Card}\left(\left(NS'_I \setminus NS_I\right) \cap \mathcal{N}\right) > 0.
\end{gathered} \tag{12}$$

Herein, a key challenge stems from the need to generate a hand-coded intrusion signature that models such attack and henceforth permits to detect it.

3.3. Signature Establishment. As mentioned before, a link spoofing attack aims to inflect the MPR selection; such

Table 2: Evidence characterizing a link spoofing attack.

Initial-evidence group	
*E*1	A MPR is replaced by a 1-hop neighbor or by an already selected MPR
*E*2	A MPR behaves maliciously, for example, it tampers, misrelays, or drops the control messages
Suspicious-evidence group	
*E*3	A MPR is the only node that provides connectivity to one (or more) 2-hops neighbor(s)
Confirmed-evidence group	
*E*4	A MPR advertises a partial set of 1-hop neighbors
*E*5	A MPR advertises nonneighboring node(s) as 1-hop neighbors(s)
Cancel-evidence group	
*E*6	A MPR is defined as an intruder/well-behaving node

selection is triggered upon a change in the symmetric 1- and 2-hops neighborhood. Rather than launching an in-depth diagnostic upon every change in the 1- or 2-hops symmetric neighborhood, we keep to a minimum the number of these diagnostics, and henceforth the amount of consumed computational power and bandwidth, by initiating it only at the occurrence of an event related to a link spoofing attack. More precisely, we ignore the changes in the 1-hop neighborhood (e.g., apparition of 1-hop neighbor) because they are observed by the node itself. Thus, they are not subject to the remote falsification which is the cornerstone of a link spoofing attack. In contrary, changes in the 2-hops neighborhood are considered as long as they impact the MPR selection. In practice, the evidences that reveal a link spoofing attack (Table 2) are broken down into:

(i) a MPR replacement (Evidence 1 or *E*1 for short) that results from a change in the covering of the 1-hop neighbors; one (or several) 1-hop neighbor(s) (possibly the replacing MPR) increase(s) its (their) coverage to the detriment of the replaced MPR. A replacement MPR is a 1-hop neighbor that is excluded from the MPRs set even though it provides the same connectivity in both the previous and current MPR selection rounds.

(ii) No MPR replacement takes place but an already selected MPR is detected as misbehaving node. For instance, a misbehaving MPR may drop, falsify, or misrelay the control messages (*E*2). Note that an evidence of a special interest herein refers to a node promoting itself as a MPR without having been selected as a MPR. A misbehaving MPR includes also the case wherein a MPR continually advertises the same 2-hops neighbors set despite being no more valid. Contrary to other cases, this one is not event driven and should be handled based on random or periodical checks;

(iii) a MPR is the only one that covers one or several nodes (*E*3);

(iv) a MPR covers partially its adjacent neighbor(s) (*E*4);

(v) a MPR provides connectivity to a nonneighboring node (*E*5).

The occurrence of either *E*1 or *E*2 constitutes the starting point of an in-depth diagnostic. *E*1 and *E*2 belong to the *initial-evidence group*. The act of being the only MPR that provides the connectivity to node(s) (*E*3) is suspicious but is not sufficient to launch an in-depth diagnostic because this situation is typical in a sparse network. Furthermore, two nodes within the covering range of each other often fail in communicating due to the unpredictable nature of wireless transmission resulting from, for example, obstacles and noises. Thus, diagnosing *E*3 is especially difficult under no specific assumption. Overall, the occurrence of *E*1 or *E*2 and optionally *E*3 leads to an in-depth diagnostic (Expression 13). In practice, the 1-hop neighbor(s) of the suspicious MPR are interrogated. Note that part of the interrogated nodes may express a different opinion that results from malicious node or an obsolete routing information. This calls for taking into account their respective reputation (as we plan in our near future work):

$$\begin{array}{cc} \multicolumn{2}{c}{(E1 \vee E2)} \\ \Downarrow & \Downarrow \\ (E4 \vee E5), \text{optional}\,(E3) & (!E4 \wedge !E5) \\ \Downarrow & \Downarrow \\ \text{The suspicious MPR} & \text{The suspicious MPR} \\ \text{is an intruder.} & \text{is well-behaving.} \end{array} \qquad (13)$$

If the obtained answers confirm (resp., infirm) that the suspicious MPR covers partially its neighbors (*E*4) and/or advertises a distant node as 1-hop neighbor (*E*5), then the MPR is declared as an intruder (resp., well-behaving) and the diagnostic is terminated (*E*6). Obviously, the cooperation between the nodes constitutes a cornerstone in our in-depth diagnostic.

3.4. Cooperative Diagnostic. The in-depth diagnostic of a link spoofing attack consists in verifying the existence of a symmetric and neighboring relationship between a suspicious MPR and its advertised 1-hop neighbors (Algorithm 1). More precisely, this diagnostic follows the following steps. First, the MPRs that have been replaced by other MPR or 1-hop neighbor are identified (lines 1-2) because such replacement represents the initial evidence of a link spoofing attack (*E*1 in Table 2). For this purpose, the function *GetReplaced-Mpr* is called (Line 1). It establishes the MPRs that have been

```
(1)  OldMprs = GetReplaced-Mpr();
(2)  SuspiciousMprs = GetReplacing-Mpr();
(3)  for (suspicious ∈ SuspiciousMprs) do
(4)      InterrogatedNodes = GetCommon2HopsNeighors
         (suspicious, OldMprs);
(5)      for (2HopsNeighbor ∈ InterrogatedNodes) do
(6)          if (LinkExistence (2HopsNeighbor, suspicious) ==
             false) then
(7)              Generate-Alarm (suspicious);
(8)          end if
(9)      end for
(10)     Cancel-Suspicious (suspicious);
(11) end for
```

Algorithm 1: In-depth diagnostic.

replaced by comparing the current MPRs and the penultimate MPRs. Then, the MPRs sharing 1-hop neighbor(s) with a replaced MPR are identified (function *GetReplacing-MPR*, line 2). Those identified MPRs are tagged as suspicious and are subject to further analysis (lines 3–11); the objective is to verify whether some spoofed links have been advertised so as to replace the MPR. It follows that the 1-hop neighbors that are common to a replaced MPR and a suspicious one are determined (*GetCommon2HopsNeighors* function, line 4). Note that these 1-hop neighbors also correspond to the 2-hops neighbors of the node launching the diagnostic. They are interrogated (*LinkExistence* function) so as to verify the existence of links. The interrogation of a 2-hops neighbor, denoted by Ai, consists in sending a request to Ai asking if this latter considers the suspicious MPR as a 1-hop neighbor at a specific time t. Whenever possible, this request is sent without going through both the suspicious MPR I or any colluding intruder I'. This eluding is necessary in order to prevent I and I' from dropping the request and/or forging a deceptive answer. Therefore, the 1-hop neighbor(s) (primarily the MPR(s)) that covers the interrogated 2-hops neighbors is provided with the request. Note that the suspecting node S is aware of is provided with the connectivity between the nodes that form its two-hops neighbors. Thus, S may select the one (if existing) that covers Ai. If no answer is obtained (i.e., the related time-out elapses), then the request is sequentially transferred through the rest of the covering 1-hop neighbors (keeping in mind that, as aforementioned, MPRs are primarily selected). However, when no neighbor is left, then an (multihops) alternative path is researched in the routing table to reach Ai. Note that the verification related to a suspicious MPR is performed within an independent thread; hence, the diagnostic of one node (and the result waiting) is not a blocking to others. If Ai denies being a 1-hop neighbor of the suspicious MPR, then a countermeasure is triggered. More precisely, an alarm that establishes the detected attacker(s) is broadcasted (*Generate-Alarm* function, line 7). Otherwise, if no deny takes place, the suspicious MPR is no longer suspected (*Cancel-Suspicious* function, line 10). In both cases, the diagnostic terminates. Note that if no answer is provided (the inconsistent answers lead also to a claim in the degree of suspicion. However, this case is not included in this paper but it is handled in our other works), then the suspicious MPR is tagged as not verified and the degree of suspicion in this latter goes up. It is worth mentioning that the current implementation of IDAR considers the link spoofing attack. However, considering other attacks is not complicated and requires only specifying the evidences to be searched in the logs and the information to be exchanged between the nodes during the in-depth diagnostic.

4. Architecture and Performance Evaluation

In this section, we introduce the key architectural components of IDAR (Section 4.1). Then, we present the evaluation of IDAR performance (Section 4.2) and further discuss the obtained result (Section 4.3).

4.1. IDAR Architecture. In order to cope with the dynamic nature of MANET, IDAR is both distributed and cooperative. The proposed architecture (Figure 2) requires that every device participates in the detection (The delegation of some detection operations (e.g., logs analyzing) may be shared among devices. But, it will not be necessarily less expensive in terms of resources consumption.) More precisely, each device contains an instance of IDAR, which independently detects the signs of suspicion and cooperates with other instances so as to conduct the diagnostic in a broader range. IDAR is implemented in *Perl* (http://www.perl.org/) and conceptually structured into 4 components.

(i) *Coordinator* that orchestrates all the components. As such, it constitutes the hearth of IDAR. In practice, it parses dynamically the OLSR logs so as to extract signs of suspicion and match these latter against predefined intrusion signatures. The *intrusion signatures* are represented as conditional rules (*if condition then state*). Furthermore, it triggers the communication and the alarm notification in order to launch advanced diagnostic.

(ii) *Communication manager* that gathers information about, for example, the adjacent links as required by the diagnostic manager. Meanwhile, it answers the diagnostic requests. This component runs into a separated thread so that other IDAR operations are not blocked.

(iii) *Knowledge database* includes the information that is extirpated from the logs and is provided by the communication manager. This database is realized in *Mysql* (http://www.mysql.com/). Stored information encompasses, for example, the 1-hop and 2-hops neighbors and the MPRs.

(iv) *Alarm notifier* is responsible for alarming the network when an intrusion is detected. Note that we are planning to include more countermeasures in the future, for example, eliminating the routes containing an intruder.

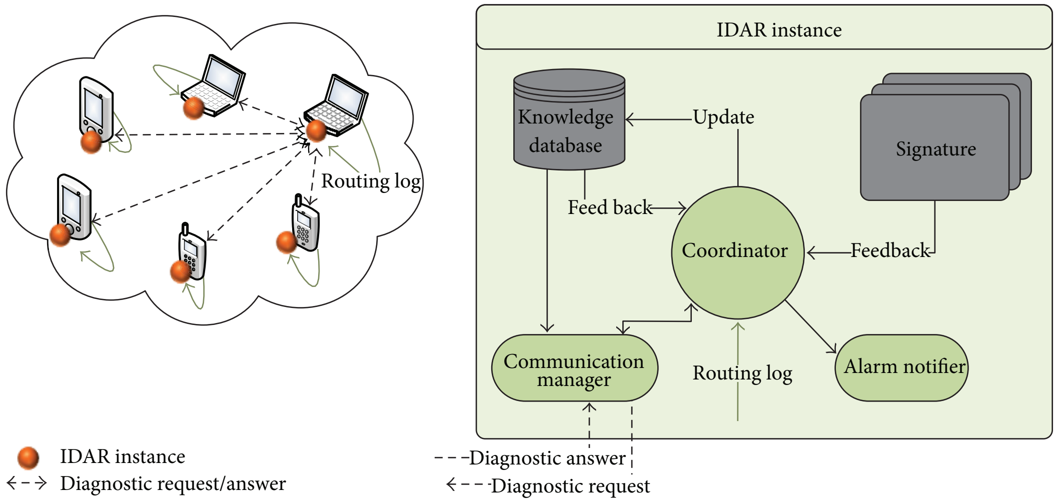


Figure 2: IDAR architecture.

All the above components have been developed so as to support intrusion detection, offering a high rate of detection along with conserving the resources.

4.2. Performance Evaluation. In order to evaluate the performance of IDAR, a mobile ad hoc network has been simulated using the network simulator NS3 (http://www.nsnam.org/) [36]. Each node in the simulated network is further coupled with a Linux Container (LXC) virtual machine (http://lxc.sourceforge.net/) [37]. The reason that motivates this coupling is twofold. First, NS3 offers the possibility to simulate a large-scale mobile ad hoc network. Second, LXC, an operating-system-level virtualization tool, permits to run multiple isolated machines (also called *containers*) on a single modified hosting kernel (up to 1024 containers over a single hosting kernel). Each container owns its proper resources (e.g., process tree, network interface, and IP address). Thus, the resource consumption (e.g., memory usage) can be isolated and measured. In practice, an instance of IDAR is installed on each container that appears as a standalone/separated machine. Figure 3 exemplifies the coupling between nodes in NS3 and LXC containers. From the IDAR perspective (as well as from any application installed on the container), the emission and reception of packets is done through the network interface (*eth0*). While the container contains a simulated ethernet card (*veth*), which is connected through an *ethernet bridge* towards a kernel tap device (*tap*). Note that for each container-node coupling, new *tap* and *ethernet bridge* should be defined in the hosting kernel. The *ethernet bridge* implements the forwarding of link-level frames. The interface *tapRouter* [38] is implemented in NS3 and is used to exchange packets between the *tap* device and the *WiFi* device, which enables NS3 nodes to communicate over a IEEE 802.11- and OLSR-enabled MANET. Overall, the outgoing packets from *eth0* in the container are delivered to the *WiFi* device at the corresponding node in the simulated network and vice versa. Simulation is carried using the aforementioned platform, wherein the hosting device is equipped with 32 GB of memory and 2 Intel(R) Xeon(R) (6) Core @2.40 GHz CPUs. Containers hold a Fedora 12 operating system. We consider a MANET (Table 3) constituted of $N = 30$ nodes split into 25 well-behaving nodes and 5 intruders. While most of other IDSs are evaluated against one intruder, we show that IDAR can handle several intruders in the same time. Intruders launch repeatedly the implemented link spoofing attack (as described in Expression 12). In our experiments, the number of successful intrusions varies between 15 and 33 according to the parameters of simulation scenario (as described in Expression 12). Nodes communicate via IEEE 802.11a and are characterized by a transmission range T_x of 90 m. They randomly move, following the so-called *RandomWalk2d* [39] mobility pattern provided by NS3: each node moves according to a randomly chosen direction at a given speed that is the same for all the nodes. When a node hits the network boundaries (unless specified, the network area is defined by a squared area of $S = 310 \times 310\,\text{m}^2$), it rebounds following a reflexive angle. Data traffic is further modeled using the *V4PingHelper* application in NS3; each node exchanges 56-byte ICMP (http://www.ietf.org/rfc/rfc792.txt) echo requests to one another and waits for 1 s before sending it again. We use OLSR as the underlying routing protocol, preserving the configuration parameters promoted in the RFC 3626 (e.g., hello message interval is 2 s).

For each experimental scenario, the simulation is launched 5 times, each one lasts for 140 s. We compare the launched attacks with the detected ones and present the average of results. We aim at evaluating the performance of IDAR in terms of:

(i) intrusion detection rate that reflects the capacity of detecting successful intrusions (a successful intrusion causes the replacement of a legitimate MPR of the victim by an intruder);

(ii) false positive rate that measures how many times a legitimate node is wrongly designated as an intruder;

TABLE 3: Simulation parameters.

Simulation platform	
Simulator	NS3 + LXC virtual machines
Container operating system	Fedora 12
Hosting device memory	32 GB
Hosting device CPU	2 Intel(R) Xeon(R) (6) Core @2.40 GHz
Simulated MANET	
Number of mobile nodes	30
Number of intruders	5
Topology	310 × 310 m^2
Transmission range	90 m
Maximum bandwidth	1 Mbps
Traffic	*V4PingHelper*: 56 bytes Icmp packets
Mobility model	*RandomWalk2d*
Maximum speed	8 m/s (or 28.8 km/h)
Simulation time	140 s

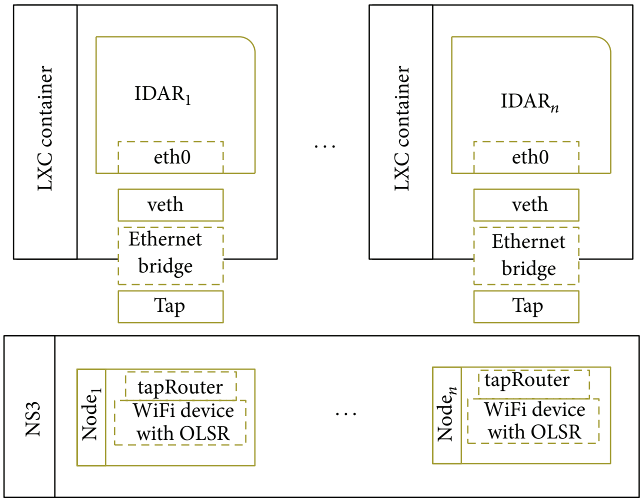


FIGURE 3: Experiment platform.

(iii) detection overhead which represents the additional network traffic that is generated because of IDAR. Note that further benchmarks are also provided in terms of memory usage.

Based on those performance indicators, IDAR is further evaluated with regard to the network density (Section 4.2.1) and node mobility (Section 4.2.2).

4.2.1. Network Density. In order to evaluate the scaling properties of IDAR, we vary the density of the network. The density of the network corresponds to the average number of neighbors, which is defined as in [40] by $(N \times \pi \times T_x^2)/S$. We selected an interval of density that ranges from 6 up to 16 neighbors, which includes the intruder, the victim, and the remaining neighbors. Figure 4(a) shows that the detection rate keeps higher than 80% regardless of the network density. This demonstrates the relatively limited impact of the network density on IDAR performance. More precisely, the intrusion detection rate slightly rises from 93.5% up to its maximum 96.3% when the density varies from 6 up to 8 neighbors: the greater is the number of neighbors, the higher is the visibility of the attack. Then, a slow diminish is observed; dense networks (with a density exceeding 8) are characterised by a high collision rate that causes a decline in the detection rate. Similarly, Figure 4(b) highlights that the average percentage of false positives is neglectful (under 5.9%). The analysis of several cases of false positive leads us to find that a false positive occurs when two nodes own a different live-times for the bidirectional link that connects these later; consequently, one of the two nodes still confirms the existence of this link while the other one denies it. To override this problem, the period during which those links are considered valid should be amplified as in [41]. Figure 4(c) presents the average memory usage caused by IDAR. This usage gently fluctuates between 17.6 MB and 22.2 MB. It is also worthy of mention that contrary to OLSR, our system does not assign additional functionality to MPR (detection may be launched by one another); IDAR provides an homogeneous load balancing among the nodes. As illustrated in Figure 4(d), the traffic caused by IDAR is very low (under 0.5% for a network density inferior to 8). It then slowly rises to reach 1.3% when the network density is 16. In order to understand the slight fall when the density reaches 14, we compare the average number of suspicious cases with regard to a varying density. This average was 70.8 (resp., 65.8) with density equals to 12 (resp., 14). Consequently, the number of diagnostic-related packets is smaller at a density of 14 than at a density of 12.

4.2.2. Mobility. In order to isolate the influence of the mobility, the network density is set to 8. The detection rate falls when the speed increases (Figure 5(a)). This results from the ever changing topology that results in increasing the number of broken links and the related drops of diagnostic-related packets. However, IDAR still provides a high detection rate; for example, the average of detection rate is 70.7%

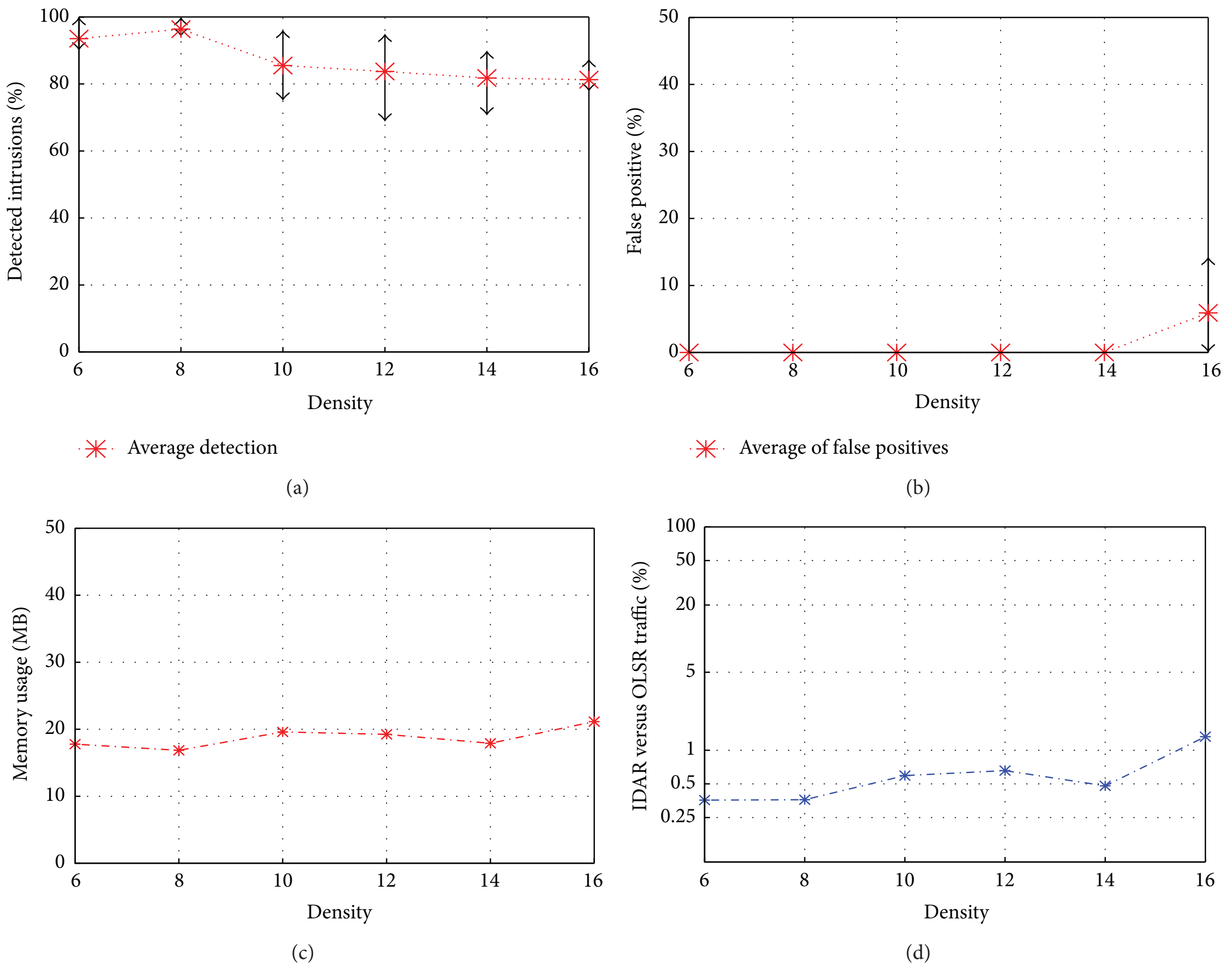


FIGURE 4: Intrusion detection rate (a), false positives rate (b), memory usage (c), and traffic (d) depending on the network density.

with a moving speed equals to 8 m/s (i.e., around 28.7 kilometers/hour equivalent to 17.89 miles/hour). Figure 5(b) shows that the higher is the moving speed, the greater it gets false positives, mainly because of out of date feedback. In fact, when a node moves away, some of its 1-hop neighbors are no longer neighbors but are still announced as such and are considered as potential attackers. A higher increase of the false positive rate is observed when the node speed ranges from 4 m/s up to 6 m/s because the instable node covering is given its transmission range, which leads to an increased update of its 1-hop neighbors. The memory usage (Figure 5(c)) rises staidly from 17.6 MB to 26.5 MB, which is reasonable even for resource-limited devices. The impact of the mobility on the bandwidth is shown in Figure 5(d). The IDAR traffic comparing to the OLSR traffic significantly grows from 0,36% up to 3.1% for a speed ranging from 0 m/s up to 2 m/s. Then, the bandwidth usage gradually grows and reaches almost 6% when the moving speed is equal to 8 m/s. This is motivated by the fact that increasing the number of verified links leads to exchange more diagnostic-related messages in the network. In order to understand the reason behind such variable growth of IDAR traffic, we counted the number of the verified suspicious neighbor links. We found that the average number of the verified links significantly grows from 45.2 to 201.4 when the velocity rises from 0 m/s to 2 m/s. This average then gradually grows and takes the values 240.8, 331.4, and 406 when the velocity equals to 4 m/s, 6 m/s, and 8 m/s, respectively. The limited number of the suspicious neighbor links when the nodes are static is justified by the fact that the attacker has a limited choice to falsify its neighbors links. Instead, when the velocity is greater than 2 m/s, there is a continuous change in the neighborhood and the attackers can attack more (diversified) targets. Meanwhile, the continuously changing topology implies an increased number of investigations.

4.3. Performance Discussion. It is clear that the density of the network holds less influence than the mobility on our IDS. More mobility in MANET causes more dropped diagnostic packets, and hence less diagnostic is concluded with a final result, that is, confirming or refuting the suspiciousness. However, comparing the launched attacks to the discovered suspicious events shows that the majority of attacks are tagged as suspicious events and considered for further diagnostic. Note that the mobility has a negative influence

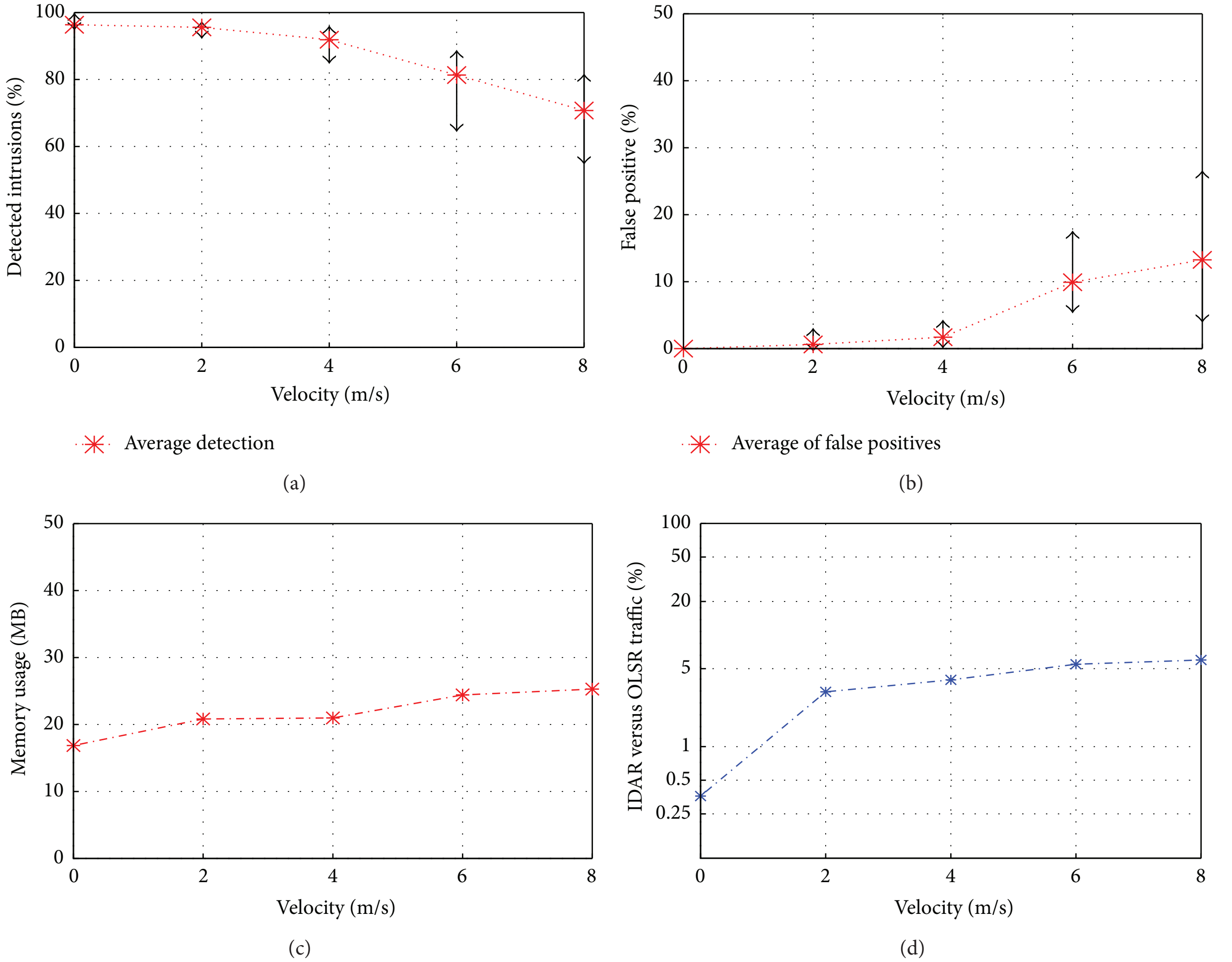


Figure 5: Intrusion detection rate (a), false positives rate (b), memory usage (c), and traffic (d) depending on the network mobility.

on all the IDSs especially the cooperative ones where there is an exchange of evidences/alerts. Table 4 provides a short performance comparing between IDAR and other known IDSs that are dedicated to ad hoc routing protocol. Like the majority of these IDSs, the performance of our system is challenged against a specific attack: link spoofing attack, in a simulated MANET (According to [42], around 75% of mobile ad hoc published papers use simulation for test issues.) Two evaluation factors are common to almost all the listed IDSs: the detection rate/capacity and the positive false rate. Even though it was challenged in a MANET wherein the intruders constitute 16.6% (Note that this percent is the 3rd highest registered percent) of the nodes, IDAR provides one of the highest detection rate. Regarding the positive false rate, we can notice that several IDSs, especially those that are dedicated to DSR protocol, show better results than our system. However, this latter still generates a limited number of false positives. It is also possible to reduce the false positives rate by amplifying the TTL (Time To Live) parameter of a bidirectional link so as to reduce the impact of obsolete neighborhood-related information during the detection. Recall that most of the generated false positives in IDAR are generated because of the difference in the time-live of a bidirectional link between the two ends of this link. In such case, this link will be expired in one end, while the other end will keep confirming its existence. It is worth mentioning that, except our system, the listed IDSs that aim to detect the attacks targeting OLSR protocol do not present neither the detection rate nor the false positives rate. But they merely prove the capacity of detecting the addressed attack(s). In addition, IDAR is one of the rare IDSs that are evaluated in terms of resources consumption. Indeed, we measure both the increment in memory usage and traffic overhead resulting from detection operations. IDAR triggers a small increment in the devices' memory usage (between 17.6 MB and 26.5 MB). Therefore, it can be tolerated by the resource-limited devices. It further triggers the lowest traffic overhead compared to the listed IDSs. IDAR presents a good scalability and offers a high detection rate even in the high density networks where a large amount of routing traffic should be monitored (Figure 4(a)). This scalability is reinforced by (i) the limited traffic overhead, (ii) the use of routing logs so as to extract the evidences instead of sniffing the traffic, (iii) the distributed nature of detection such that each node is responsible for detecting the attacks targeting it, and (iv) the unnecessity of modifying the used routing protocol.

5. Related Work

Systems that detect intrusion targeting ad hoc routing protocol are extremely diverse in the way they analyze intrusion. They fall into the three key categories.

(i) *Anomaly detection* system defines the correct behavior of the node/network so as to detect deviations to this behavior. This correct behavior is automatically built during an attackless training phase. The detection accuracy depends on the ability to (i) describe the correct behavior and (ii) distinguish between anomalous and unexpected behaviors.

(ii) *Specification-based detection* system hand-codes the legitimate function/operation and then searches for a violation to this operation.

(iii) *Signature-based detection* system first describes the way an intruder penetrates the system by defining an intrusion signature. Then, any behavior that is close to this predefined signature is flagged as intrusion.

Hereafter, we detail examples of each of these categories. Anomaly-based detection constitutes the main approach used to detect attacks. The IDS proposed by Zhang et al. [7] constitutes a de facto standard in MANET. It aims at detecting the attempts to falsify the routes provided by the AODV [61], DSR [62], and DSDV [63] routing protocols. During the training phase, the impact of movement on the percentage of changes in the routing table is analyzed. Note that this movement (i.e., velocity, direction, and position) is provided by a Global Positioning System (GPS). Then, during the operation phase, an actual percentage of changes differing from the predicted one, is defined as anomaly. In practice, the distinguishing between anomalous and normal behaviors is provided by the Support Vector Machine (SVM) Light [64] classifier or by RIPPER [15], a rule-based engine. The hierarchical IDS proposed in [47] detects *blackhole* and routing request flooding attacks targeting AODV protocol. Here, MANET is arranged into clusters, whereas the node with the highest residual energy and number of connections is elected as a cluster-head. Each cluster-head monitors the traffic inside its cluster so as to identify the abnormal values of (i) the percentage of changes in the routing table and (ii) the propagation of routing/data packets. Such abnormalities are identified thanks to (1-SVM) classifier [65], a deviation of SVM that needs to be trained with either normal or abnormal scenario but not both of them. In [48] (resp., [66]), a blackhole (and resp., dropping) attack targeting AODV protocol (resp., a secured version of AODV protocol including authentication and reputation estimation) are detected by investigating features, for example, the number of route requests and route replies as well as the average difference of sequence numbers (Largely increased sequence numbers are known as a sign of blackhole attack). Then, simple anomaly detection is applied: if the distance between the actually observed features and the average ones that are recorded during the training exceeds a given threshold, then an intrusion is detected. More sophisticated cross-features analysis (CFA) [67] is applied to detect both blackhole and packet dropping on AODV and DSR protocols. Features including, for example, the reachability between 2 nodes and the number of delivered packets, are analyzed within time windows. This analysis attempts to quantify the relation existing between one feature f_i and the others $f_1, \ldots, f_{i-1}, f_{i+1}, \ldots, f_k$ (with $k + 1$ defining the number of features analyzed). During the operation phase, the probability of matching between the observed feature f_i' and the predicted feature f_i, which is established based on $f_1', \ldots, f_{i-1}', f_{i+1}', \ldots, f_k'$, is calculated by a decision tree classifier named C4.5 [68]. An anomaly refers to an average of matching probability fewer than a given threshold. CFA and C4.5 are also used in [69] with traffic-related features (e.g., number of received/transferred packets) and OLSR- or AODV-related features (e.g., MPR updates for the former and route discovery for the latter). A modified Markov chain-based IDS is proposed in [58] so as to detect route disruption attack targeting DSR protocol. Here, the detection depends on identifying unpredicted value of two features: the percentage of changes in the number of routes and the percentage of changes in the number of hops. In practice, a Markov chain is based on the immediate previous n consecutive values of a feature as to predict the $(n + 1)$th value. If the difference between a predicted and an observed value exceeds a given threshold, an intrusion is concluded.

Rather than establishing automatically a correct behavior, specification-based detection system hand-codes the correct behavior of the routing protocols based on the protocol specification (IETF draft or RFC). Then, the system attempts to detect a violation of constraints circumventing this behavior. Four key constraints define the correct behavior of OLSR ([14, 70]).

(1) Neighbor relation must be reciprocal (i.e., 2 neighbors must hear the hello message sent by each other).

(2) The MPRs and the nodes that select the MPR (i.e., the MPR selectors) must be adjacent.

(3) A node that finds itself advertised as a MPR selector in a TC message must be adjacent to the originator of this message.

(4) Nodes receive TC messages without modifications from MPR(s).

Each node sniffs the traffic in its radio range so as to discover the violations of any of the above constraints. These constraints are modeled by semantic properties in [70] (resp., rules in [14]) so as to detect link spoofing attack on TC (and resp., hello messages); both experiment an ability to detect a link spoofing by observing a violation of the third condition over a simulated network composed of 11 nodes. Finite-state machines are used to describe similar constraints on OLSR [41] and the behavior of AODV protocol [52, 71]. In [41], a centralized detection of link spoofing, man-in-the-middle, and deny of service attack, is performed. While in [52, 71], incorrect hop counts/sequence numbers are analyzed by distributed sensors, which sniff and group the packets per request-reply flow so as to estimate the forwarding path per flow. If a route request/reply is illegally modified or forwarded via a nonexpected path, then an alarm is triggered. In [60], similar FSMs are used to define constraints on, both, the route discovery in AODV and the packet forwarding operations.

Table 4: Comparing IDAR performance with other IDSs.

IDS	Routing protocol	Number of nodes	Intruders (%)	Moving speed	Detection rate	False positives rate	Increment in memory usage	Increment in traffic overhead
IDAR	OlSR	30	16.6%	0–8 m/s	70.7%–96.3%	0%–13.2%	17.6–26.5 MB	47.8–621.5 KB (0.36%–5.97% of OlSR traffic)
[43]	OlSR	30	3.3%, 6.6%, 10%	—	—	—	—	6%–12% of OlSR traffic
[41]	OlSR	10	10%	0 m/s	Detection is possible	—	—	—
[44]	OlSR	7	14.3%	0 m/s	Detection is possible	—	—	—
[45]	OlSR	5	20%	0 m/s	Detection is possible	—	—	—
[14]	OlSR	11	9%	0 m/s	Detection is possible	—	—	—
[46]	AODV	30	3.3%	0–5 m/s	30.67%–90%	0%–20%	—	—
[47]	AODV	50	2%	0–20 m/s	90%	5%	—	—
[48]	AODV	30	3.3%	1–20 m/s	70%–82%	12%–18%	—	—
[49]	AODV	20, 50	10%	1–10 m/s	96.4%–98.7%	0.79%–0.93%	—	900 kB
[50]	AODV	52	3.8%	0–20 m/s (Fixed IDSs)	100%	0%–0.6%	—	—
[51]	AODV	5, 21	20%, 23.8%	0–10 m/s	50%–100%	0%–30%	—	—
[52]	AODV	30	—	0–20 m/s	70%–100%	0.5%–15%	—	—
[53]	AODV	50	Random	0–20 m/s	79 ± 10%–92 ± 3%	5 ± 1%–32 ± 8%	—	—
[54]	AODV	10, 20, 50	—	0–10 m/s	0%–100%	0%– ≫ 100%	0.2%–0.6%	—
[55]	AODV	20–200	—	0–20 m/s	30%–95%	0%–16%	—	—
[56]	AODV	50	—	1–20 m/s	99.8%–100%	0.83%–8.46%	—	—
[7]	AODV	—	—	—	88.48 ± 4.14%–97.1 ± 0.32%	1.45±0.72%–20.2±6.27%	—	—
	DSDV	—	—	—	85.23±3.28%–90.61±2.99%	5.37±3.10%–26.3±5.49%	—	—
	DSR	—	—	—	85.2 ± 2.38%–99.1 ± 0.37%	0.03±0.04%–15.3±4.08%	—	—
[57]	DSR	50	40%	0–20 m/s	—	—	—	11%–31.3% of data traffic
[58]	DSR	30	3.3%	3–5 m/s	60%–100%	2.5%–50%	—	—
[59]	DSR	30	—	—	75%	1.2%	—	—
[60]	DSR	50	2%	0–20 m/s	83.7%–97.4%	1.3%–7.2%	—	—

A signature-based system distinguishes itself by modeling the misbehavior (rather than the correct behavior) so as to identify if a sequence of observed events matching an intrusion signature. AODVSTAT [54] uses state-based signatures in order to detect dropping and spoofing attacks along with network flooding. Few sensors sniff the traffic and match it against predefined signatures. They also exchange periodically MAC and IP addresses so as to detect an identity spoofing which is characterized by a node emitting a packet identified by MAC and IP addresses differing from those registered for this node. A dropping attack refers to a node that fails to replay route request/reply. Finally, a node sending a number of packets exceeding a given threshold is identified as willing to exhaust resources. In [45], intrusion signatures are specified in opposition to the legitimate behavior of OLSR depicted in conjunction with specification rules similar to those defined in [70]. In particular, a node N mistrusts two neighbors L and M that conform to the following rules: M advertises L as a neighbor while L does not advertise M as a neighbor (or reciprocally), both L and M send TC messages while L's neighbors are a subset of M's neighbors (or reciprocally). In addition, N mistrusts M if N which is not a neighbor of M finds itself as a MPR selector in a TC of N (constraint 3). In [44], a FSM is employed to model the signature of hello message fabrication in OLSR protocol in an agent-based IDS. In practice, each node uses a Simple Network Management Protocol (SNMP) agent so as to collect audit data from the Management Information Base (MIB). After that, events are extracted from the collected audit data and are matched to the attack signature. The addressed attack aims at breaking the link between a victim node and its neighbors, and thus, a DoS takes place. To that end, the attacker impersonates the identity of the victim and sends a fake hello message advertising one of the victim symmetric neighbors with lost link status. Upon receiving the fabricated message, the neighbor changes the status of its link with the victim to "heard" and stops routing packets through the victim.

5.1. Synthesis and Discussion. A great majority of the literature is focused on anomaly detection while scarce effort investigates specification- and signature-based intrusion. This naturally calls for consolidating the effort on specification- and signature-based detection while following the *habitus* of wired network which consists in coupling these detection systems with one another, for example, in [72] wherein a combination of anomaly- and signature-based detection is realized so as to increase the detection rate. Almost all of the IDS focuses only on the detection accuracy, that is, providing a high detection rate along with limited false alarms. However, they do not consider the criticality of maintaining the resources and henceforth extending the lifetime of the node/network. Our IDS takes into account the necessity of reducing to minimum, both the communication and computation overload related to intrusion identification. The absolute majority of the IDSs is simulated (including the system we are proposing) rather than being either developed or experimented on testbeds or relying on empirical data. The reason that explains this situation is twofold. First, the deployment of critical MANET is limited/not advertised. Second, to the best of our knowledge, experiences of intrusion in MANETs are not reported: intrusions are in fact developed/simulated as proofs of concept. Nevertheless, not only routing protocols but also the characterization of the intrusion together leverage on the experience gained in designing routing protocols for wired networks and dealing with related misbehaviors. In particular, the similarity existing between the OSPF and OLSR protocols implies that many threats (e.g., sequence numbers, identity spoofing) are shared. In other words, intrusion may be inspired from the one targeting wired networks. Meanwhile, the development of intrusion detection and the envisioning of attacks are accelerated.

6. Conclusion

We survey and classify the attacks that target ad hoc routing protocols focusing on the OLSR protocol. In order to facilitate the definition of intrusion signatures, we extend a description model—an attack is expressed as the preconditions and the resulting consequences—and enrich it with temporal annotations. Once hand-coded, these signatures are utilized by IDAR, a log-based, distributed intrusion detection system dedicated to operate in mobile ad hoc networks. IDAR distinguishes itself by analyzing the logs generated by a routing protocol and extracts intrusion evidences so as to compare these latter against predefined intrusion signatures. For this purpose, evidences are categorized into four groups according to their degree of suspicion/gravity and hence to their ability to activate/deactivate the diagnostic. We further develop a link spoofing attack on the OLSR protocol, build the related detection rules, and evaluate the performances of IDAR relying on the NS3 simulator coupled with LXC virtual machines. Overall, the experiments figure out a high rate of intrusion detection and low false positives rate even under increased mobility and density. Meanwhile, resource consumption and network overhead result from the diagnostic are low and adapted to the resource-constrained devices. Note that ongoing effort is provided as to evaluate IDAR in a real MANET consisting of resource-constrained devices. Still, detecting intrusion is not a trivial task due to the large number of evidences to tackle and the resource-consuming diagnostic. A compromise between detection accuracy and resource consumption could be found. For this purpose, we are working on a new statistical mechanism for gathering the evidences. This mechanism aims at enhancing the scalability of our system by restricting the interrogation to a limited subset of nodes during the diagnostic. Moreover, until now, we assume that honest parties are passive; they alarm others but do not react. We are considering the coupling with countermeasures (e.g., blacklisting) and the exploring of lightweight binary consensus and trust establishment among the nodes that participate in the intrusion detection.

References

[1] Z. Zhao, H. Hu, G. Ahn, and R. Wu, "Risk-aware mitigation for manet routing attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 2, pp. 250–260, 2012.

[2] B. Sun, L. Osborne, Y. Xiao, and S. Guizani, "Intrusion detection techniques in mobile ad hoc and wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 5, pp. 56–63, 2007.

[3] S. Sesay, Z. Yang, and J. He, "A survey on mobile ad hoc wireless network," *Information Technology Journal*, vol. 3, no. 2, pp. 168–175, 2004.

[4] P. Brutch and C. Ko, "Challenges in intrusion detection for wireless ad-hoc networks," in *Proceedings of the Symposium on Applications and the Internet Workshops (SAINT '03)*, pp. 368–373, 2003.

[5] S. Khan, N. Alrajeh, and K. Loo, "Secure route selection in wireless mesh networks," *Computer Networks*, vol. 56, no. 2, pp. 491–503, 2012.

[6] S. Corson and J. Macker, "Mobile ad hoc networking (manet): routing protocol performance issues and evaluation considerations," IETF RFC2501, 1999.

[7] Y. Zhang, W. Lee, and Y. Huang, "Intrusion detection techniques for mobile wireless networks," *Wireless Networks*, vol. 9, no. 5, pp. 545–556, 2003.

[8] B. C. Cheng and R. Y. Tseng, "A context adaptive intrusion detection system for MANET," *Computer Communications*, vol. 34, no. 3, pp. 310–318, 2011.

[9] T. Eissa, S. A. Razak, and M. D. A. Ngadi, "Towards providing a new lightweight authentication and encryption scheme for MANET," *Wireless Networks*, vol. 17, no. 4, pp. 833–842, 2011.

[10] A. Irshad, W. Noshairwan, M. Shafiq, S. Khurram, E. Irshad, and M. Usman, "Security enhancement for authentication of nodes in MANET by checking the CRL status of servers," *Communications in Computer and Information Science*, vol. 78, pp. 86–95, 2010.

[11] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. 13, no. 2, pp. 222–232, 1987.

[12] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," IETF Experimental RFC 3626, 2003.

[13] P. Albers, O. Camp, J. Percher et al., "Security in ad hoc networks a general intrusion detection architecture enhancing trust based approaches," 2002.

[14] F. Cuppens, N. Cuppens-Boulahia, S. Nuon, and T. Ramard, "Property based intrusion detection to secure OLSR," in *Proceedings of the 3rd International Conference on Wireless and Mobile Communications (ICWMC '07)*, Guadeloupe, France, March 2007.

[15] W. Cohen, "Fast effective rule induction," in *Proceedings of the International Conference on Machine Learning (ICML '95)*, 1995.

[16] S. Sarafijanovic and J.-Y. le Boudec, "An artificial immune system for misbehavior detection in mobile ad-hoc networks with virtual thymus, clustering, danger signal and memory detectors," in *Proceedings of the 3rd International Conference on Artificial Immune Systems (ICARIS '04)*, 2004.

[17] S. Sen and J. Clark, *Intrusion Detection in Mobile Ad Hoc Networks in Guide to Wireless Ad Hoc Networks*, Springer, London, UK, 2009.

[18] D. Subhadrabandhu, S. Sarkar, and F. Anjum, "A Framework for misuse detection in ad hoc networks—part II," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 290–303, 2006.

[19] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1548–1557, April 2001.

[20] A. P. Lauf, R. A. Peters, and W. H. Robinson, "A distributed intrusion detection system for resource-constrained devices in ad-hoc networks," *Ad Hoc Networks*, vol. 8, no. 3, pp. 253–266, 2010.

[21] N. A. Alrajeh, S. Khan, J. Lloret, and J. Loo, "Secure routing protocol using cross-layer design and energy harvesting in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 374796, 11 pages, 2013.

[22] S. Sen, J. Clark, and J. Tapiador, *Security Threats in Mobile Ad Hoc Networks, I. S. of Self-Organizing Networks*, Auerbach Publications, 2010.

[23] P. Ning and K. Sun, "How to misuse AODV: a case study of insider attacks against mobile ad-hoc routing protocols," *Ad Hoc Networks*, vol. 3, no. 6, pp. 795–819, 2005.

[24] T. Clausen, C. Dearlove, P. Jacquet, and U. Herberg, "The optimized link state routing protocol version 2," IETF Active Internet-Draft Draft-Ietf-Manet-Olsrv2-15, 2012.

[25] U. Herberg and T. Clausen, "Security issues in the optimized link state routing protocol version 2 (olsrv2)," *International Journal of Network Security & Its Applications*, vol. 2, no. 2, 2010.

[26] Y. Owada, T. Maeno, H. Imai, and K. Mase, "OLSRv2 implementation and performance evaluation with link layer feedback," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '07)*, pp. 67–72, New york, NY, USA, August 2007.

[27] F. Hong, L. Hong, and C. Fu, "Secure OLSR," in *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA '05)*, pp. 713–718, March 2005.

[28] L. N. A. M. Hegland, P. Spilling, and Q. Kure, "Hybrid protection of olsr," in *Electronic Notes in Theoretical Computer Science*, 2006.

[29] P. Kunwar, S. Sofat, and D. Bansal, "Comparison of secure olsr routing protocol," *International Journal of Engineering Science*, vol. 3, no. 6, p. 5049, 2011.

[30] A. Adnane, C. Bidan, and R. T. de Sousa Junior, "Trust-based countermeasures for securing OLSR protocol," in *Proceedings of the 7th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC '09)*, pp. 745–752, August 2009.

[31] B. Wu, J. Chen, J. Chen, J. Wu, and M. Cardei, "A survey of attacks and countermeasures in mobile ad hoc networks," in *Wireless Network Security*, Springer, New York, NY, USA, 2007.

[32] C. Adjih, D. Raffo, and P. Mühlethaler, "Attacks against OLSR: distributed key management for security," in *Proceedings of the OLSR Interop Workshop*, 2005.

[33] G. Cervera, M. Barbeau, J. Garcia-Alfaro, and E. Kranakis, "Mitigation of topology control traffic attacks in OLSR networks," in *Proceedings of the 5th International Conference on Risks and Security of Internet and Systems (CRiSIS '10)*, October 2010.

[34] P. M. Jawandhiya, M. M. Ghonge, M. S. Ali et al., "A survey of mobile ad hoc network attacks," *International Journal of Engineering Science and Technology*, vol. 2, no. 9, pp. 4063–4071, 2010.

[35] L. M. Feeney, "An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 6, no. 3, pp. 239–249, 2001.

[36] G. Riley and T. Henderson, "The ns-3 network simulator," in *Modeling and Tools for Network Simulation*, Springer, Berlin, Germany, 2010.

[37] S. Bhattiprolu, E. Biederman, S. Hallyn et al., "Virtual servers and checkpoint/restart in mainstream linux," *SIGOPS Operating Systems Review*, vol. 42, no. 5, pp. 104–113, 2008.

[38] J. Zhang, J. Xing, and Z. Qin, "Taprouter: an emulating framework to run real applications on simulated mobile ad hoc network," in *Proceedings of the 44th Annual Simulation Symposium (ANSS '11)*, pp. 39–46, 2011.

[39] D. Broyles, A. Jabbar, and J. P. G. Sterbenz, "Design and analysis of a 3-d gauss-markov mobility model for highly-dynamic airborne networks," in *Proceedings of the International Telemetering Conference (ITC '10)*, San Diego, CA, USA, 2010.

[40] J. Haerri, F. Filali, and C. Bonnet, "Performance comparison of AODV and OLSR in vanets urban environments under realistic mobility patterns," in *Proceedings of the IFIP Med-Hoc-Net Workshop*, 2006.

[41] C. H. Tseng, T. Song, P. Balasubramanyam, C. Ko, and K. Levitt, "A specification-based intrusion detection model for OLSR," *Recent Advances in Intrusion Detection*, vol. 3858, pp. 330–350, 2006.

[42] S. Kurkowski, T. Camp, and M. Colagrosso, "Manet simulation studies: the incredibles," *SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 4, pp. 50–561, 2005.

[43] A. Abdalla, I. Saroit, A. Kotb et al., "Misbehavior nodes detection and isolation for MANETs OLSR protocol," *Procedia Computer Science*, vol. 3, pp. 115–121, 2011.

[44] R. Puttini, J. Percher, L. Me et al., "A modular architecture for distributed ids in manet," in *Proceedings of the International Conference on Computational Science and Its Applications III*, 2003.

[45] A. Adnane, R. T. de Sousa, L. Mé, and C. Bidan, "Autonomic trust reasoning enables misbehavior detection in OLSR," in *Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC '08)*, pp. 2006–2013, March 2008.

[46] W. Wang, H. Man, and Y. Liu, "A framework for intrusion detection systems by social network analysis methods in ad hoc networks," *Security and Communication Networks*, vol. 2, no. 6, pp. 669–685, 2009.

[47] H. Deng, R. Xu, J. Li, F. Zhang, R. Levy, and W. Lee, "Agent-based cooperative anomaly detection for wireless ad hoc networks," in *Proceedings of the 12th International Conference on Parallel and Distributed Systems (ICPADS '06)*, pp. 613–620, July 2006.

[48] S. Kurosawa, H. Nakayama, N. Kato et al., "Detecting blackholes attack on AODV-based mobile ad hoc networks by dynamic learning method," *International Journal of Network Security*, vol. 5, no. 3, pp. 338–346, 2007.

[49] W. Wang, H. Wang, B. Wang et al., "Energy-aware and selfadaptive anomaly detection scheme based on network tomography in mobile ad hoc networks," *Information Sciences*, vol. 220, pp. 580–602, 2013.

[50] M.-Y. Su, "Prevention of selective black hole attacks on mobile ad hoc networks through intrusion detection systems," *Computer Communications*, vol. 34, no. 1, pp. 107–117, 2011.

[51] N. Marchang and R. Datta, "Collaborative techniques for intrusion detection in mobile ad-hoc networks," *Ad Hoc Networks*, vol. 6, no. 4, pp. 508–523, 2008.

[52] B. V. R. N. Yadav, B. Satyanarayana, and O. B. V. Ramanaiah, "An efficient intrusion detection system for mobile ad hoc networks," in *Emerging Trends in Computing, Informatics, Systems Sciences, and Engineering*, vol. 151, Springer, New York, NY, USA, 2013.

[53] Y. Huang and W. Lee, "Attack analysis and detection for ad hoc routing protocols," in *Recent Advances in Intrusion Detection*, vol. 3224, Springer, Berlin, Germany, 2004.

[54] G. Vigna, S. Gwalani, K. Srinivasan, E. M. Belding-Royer, and R. A. Kemmerer, "An intrusion detection tool for AODV-based ad hoc wireless networks," in *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC '04)*, pp. 16–27, December 2004.

[55] M. Ayachi, C. Bidan, and N. Prigent, "A trust-based ids for the aodv protocol," in *Information and Communications Security*, vol. 6476, pp. 430–444, 2010.

[56] S. Şen and J. A. Clark, "A grammatical evolution approach to intrusion detection on mobile ad hoc networks," in *Proceedings of the 2nd ACM Conference on Wireless Network Security (WiSec '09)*, pp. 95–102, March 2009.

[57] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 255–265, August 2000.

[58] B. Sun, K. Wu, and U. Pooch, "Routing anomaly detection in mobile ad hoc networks," in *Proceedings of The 12th International Conference on Computer Communications and Networks (ICCCN '03)*, 2003.

[59] S. Bose, S. Bharathimurugan, and A. Kannan, "Multi-layer integrated anomaly intrusion detection system for mobile adhoc networks," in *Proceedings of the International Conference on Signal Processing, Communications and Networking (ICSCN '07)*, pp. 360–365, February 2007.

[60] P. Yi, Y. Zhong, and S. Zhang, "A novel intrusion detection method for mobile ad hoc networks," in *Advances in Grid Computing*, vol. 3470, pp. 1183–1192, Springer, Berlin, Germany, 2005.

[61] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," IETF Experimental RFC 3561, 2003.

[62] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, pp. 153–181, Kluwer Academic, 1996.

[63] C. Perkins and P. Bhagwat, "Highly dynamic destinationsequenced distance-vector routing (dsdv) for mobile computers," in *Proceedings of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94)*, vol. 24, pp. 234–244, 1994.

[64] T. Joachims, *Making Large-Scale Support Vector Machine Learning Practical*, MIT Press, Cambridge, Mass, USA, 1999.

[65] Y. Yajima and T. Kuo, "Efficient formulations for 1-svm and their application to recommendation tasks," *Journal of Computers*, vol. 1, no. 3, pp. 27–34, 2006.

[66] L. Bononi and C. Tacconi, "Intrusion detection for secure clustering and routing in Mobile Multi-hop Wireless Networks," *International Journal of Information Security*, vol. 6, no. 6, pp. 379–392, 2007.

[67] Y. Huang, W. Fan, W. Lee et al., "Cross-feature analysis for detecting ad-hoc routing anomalies," in *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS '03)*, pp. 478–487, May 2003.

[68] J. Quinlan, *C4. 5: Programs for Machine Learning*, Morgan Kaufmann, Boston, Mass, USA, 1993.

[69] J. B. D. Cabrera, C. Gutiérrez, and R. K. Mehra, “Ensemble methods for anomaly detection and distributed intrusion detection in Mobile Ad-Hoc Networks,” *Information Fusion*, vol. 9, no. 1, pp. 96–119, 2008.

[70] M. Wang, L. Lamont, P. Mason, and M. Gorlatova, “An effective intrusion detection approach for OLSR MANET protocol,” in *Proceedings of the 13th IEEE International Conference on Network Protocols Workshop (ICNP ’05)*, pp. 55–60, November 2005.

[71] C. Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt, “A specification-based intrusion detection system for AODV,” in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 125–134, October 2003.

[72] S. A. Razak, S. M. Furnell, N. L. Clarke, and P. J. Brooke, “Friend-assisted intrusion detection and response mechanisms for mobile ad hoc networks,” *Ad Hoc Networks*, vol. 6, no. 7, pp. 1151–1167, 2008.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 815959, 8 pages
http://dx.doi.org/10.1155/2013/815959

Research Article

PUE Attack Detection in CWSN Using Collaboration and Learning Behavior

Javier Blesa, Elena Romero, Alba Rozas, Alvaro Araujo, and Octavio Nieto-Taladriz

Electronic Engineering Department, ETSI Telecomunicación, Universidad Politécnica de Madrid, 28040 Madrid, Spain

Correspondence should be addressed to Javier Blesa; jblesa@die.upm.es

Received 1 March 2013; Accepted 3 June 2013

Academic Editor: S. Khan

Cognitive Wireless Sensor Network (CWSN) is a new paradigm which integrates cognitive features in traditional Wireless Sensor Networks (WSNs) to mitigate important problems such as spectrum occupancy. Security in Cognitive Wireless Sensor Networks is an important problem because these kinds of networks manage critical applications and data. Moreover, the specific constraints of WSN make the problem even more critical. However, effective solutions have not been implemented yet. Among the specific attacks derived from new cognitive features, the one most studied is the Primary User Emulation (PUE) attack. This paper discusses a new approach, based on anomaly behavior detection and collaboration, to detect the PUE attack in CWSN scenarios. A nonparametric CUSUM algorithm, suitable for low resource networks like CWSN, has been used in this work. The algorithm has been tested using a cognitive simulator that brings important results in this area. For example, the result shows that the number of collaborative nodes is the most important parameter in order to improve the PUE attack detection rates. If the 20% of the nodes collaborates, the PUE detection reaches the 98% with less than 1% of false positives.

1. Introduction

One of the fastest growing sectors in recent years has undoubtedly been that of WSNs. WSNs consist of spatially distributed autonomous sensors that monitor a wide range of ambient conditions and cooperate to share data across the network. WSNs are increasingly being introduced into our daily lives. Potential fields of applications can be found, ranging from the military to home control commercially or industrially, to name a few. The emergence of new wireless technologies such as ZigBee and IEEE 802.15.4 has allowed for the development of interoperability among commercial products, which is important for ensuring scalability and low cost. Most WSN solutions operate on unlicensed frequency bands. In general, they use industrial, scientific, and medical (ISM) bands, like the worldwide available 2.4 GHz band. This band is also used by a large number of popular wireless applications, for example, those that work over Wi-Fi or Bluetooth. For this reason, the unlicensed spectrum bands are becoming overcrowded. As a result, coexistence issues on unlicensed bands have been the subject of extensive research, and, in particular, it has been shown that IEEE 802.11 networks can significantly degrade the performance of ZigBee/802.15.4 networks when operating on overlapping frequency bands [1].

The increasing demand for wireless communication presents a challenge to make efficient use of the spectrum. To address this challenge, cognitive radio (CR) has emerged as the key technology, which enables opportunistic access to the spectrum. A CR is an intelligent wireless communication system that is aware of its surrounding environment and adapts its internal parameters to achieve reliable and efficient communication. These new networks have many applications, such as the cognitive use of the TV white space spectrum or making secure calls in emergency situations. In order to create these new applications, CR differentiates between two kinds of users; primary users (PUs) are licensed users, and secondary users (SUs) are those who try to use the same bands when they detect a spectral hole. Adding cognition to the existing WSN infrastructure brings about a lot of benefits. However, cognitive technology will not only provide access to new spectrum bands but will also provide better propagation characteristics. By adaptively changing system parameters like modulation schemes, transmit power, carrier frequency, and constellation size, a wide variety of data rates can be achieved. This will certainly improve power consumption, network life, and reliability in a WSN.

The nature of large, dynamic, adaptive, and Cognitive Wireless Sensor Networks presents significant challenges in designing security schemes. A Cognitive Wireless Sensor Network is a special network that has many constraints and many different features compared to traditional WSNs. While security challenges have been widely tackled in traditional networks, it is a novel area in Cognitive Wireless Sensor Networks. The wireless medium is inherently less secure than the wired one because its broadcast nature makes eavesdropping simple. Any transmission can be easily intercepted, altered, or replayed by an adversary. The wireless medium allows an attacker to easily intercept valid packets and easily inject malicious ones. Cognitive features allow for a dynamic reconfiguration to avoid these attacks. However, malicious nodes can use the dynamic reconfiguration to create new attacks such as Primary User Emulation (PUE). PUE is a new attack where a malicious node emulates the behavior of an incumbent node with the purpose of using the radio spectrum for its own interest or denying the access to other nodes.

To avoid these kinds of attacks some approaches have been investigated such as location-based approaches, but other cognitive features such as collaboration and learning have not been sufficiently exploited. We need to take into account that most WSNs have been developed in order to carry out a specific application. As a result, nodes usually have their own behavior pattern. This characteristic gives the network an opportunity to create a node profile for each sensor. These profiles can be created and optimized thanks to cognitive features such as spectrum awareness, learning, and collaboration. In this paper, simulations show how collaboration is essential to improve detection. Moreover, collaboration is the parameter that contributes most efficiently. The node profiles are used to detect anomalies in behavior and, for instance, PUE attacks.

The increasing use of WSN in many critical applications represents an important risk and a motivation for the study of the PUE attack. If a WSN that monitors a factory is attacked, the possibilities of errors in the systems increase. This means loss of money and replacement of machines. Another example is the home security systems. The malicious nodes could send corrupt information in order to hack the access service. If the system can detect the attack, it will omit the information from the attacker.

The organization of this paper is as follows. Section 2 explains the specific characteristics of the CWSN scenarios related to this work. In Section 3, works in security for PUE attacks is reviewed. In Section 4 a brief introduction to the main topics related to behavior learning and security are provided. Then, in Section 5, assumptions taken in account for the simulations are specified. Section 6 explains the general architecture of the system, while Section 7 provides its evaluation. Finally, the conclusions are shown in Section 8.

2. Cognitive Wireless Sensor Networks

A CWSN scenario includes multiple wireless sensor nodes, usually with a specific application. There are some specific characteristics of these CWSNs that imply some changes on how we understand these cognitive networks.

For example, CWSNs usually operate in the ISM bands, where anyone can transmit without license. Because of this feature, the definition of primary users (PUs) and secondary users (SUs) should be different. For this CWSNs definition, the differences between PUs and SUs are based on the priority of their functionality. For example, a fire sensor would be of more priority than a temperature sensor. In our case, an SU only transmits the prepare information when no PU is transmitting.

3. Related Work

According to Section 1 it is very clear that CWSNs face a dangerous problem in security. Several attacks could be adapted from WSNs to the new paradigm of cognitive networks. In the last ten years some researches related to security on CRNs have appeared. They describe specific attacks against these networks, but few countermeasures are proposed.

Most of the studies in security are focused on PUE detection. According to the origin of cognitive radio networks, the efficient use of TV spectrum in the USA and early studies used the location in order to detect malicious attacks. These PUs are TV towers with a precise behavior and location.

In [2] Chen and Park present the first method to detect a PUE attack based on location. The idea of this method is to differentiate the attacker from a licensed user comparing the transmission origin with the previously known PU position.

The same authors use a mechanism based on location in [3]. Moreover, they include some new parameters, such as the signal as power or RF fingerprints, to decide the nature of the signal.

In [4] the authors assume that the attacker is close to the victim and the real PU is much farther from the SUs than the attacker. Moreover, the position of each node, including the attacker, is fixed. Assuming that SUs can learn about the characteristics of the spectrum according to the received power, the authors in [5] follow a similar approach. Although they do not use any location information, they assume a static scenario with the PU much farther away from other possible malicious nodes than the SUs.

More location-based countermeasures can be found in [6, 7]. In the first work, secondary users calculate the estimated position of the PUE and then propagate this knowledge to carry out a coordinate decision. The second work is focused on the algorithm to detect the position of the PUE.

All these countermeasures are only based on the location. This characteristic cannot be used in some CWSN scenarios where both SUs and PUs can be mobile. Therefore, it is very clear that another approach should be adopted.

A few different solutions, not based on location, have been presented. In [8] the authors use the phase noise of a local oscillator as a fingerprint to identify the incumbent signals from the attacking ones.

Finally, in [9] the authors present a differential game approach to mitigate the PUE attack. Based on the assumption that PUE attacker has less energy than the PUs, they look for the optimal sensing strategy of SU. The Nash equilibrium solution is obtained.

Although these two last approaches are valid for mobile PUE attackers in CWSN, the algorithms implemented require relatively high computational resources, which is an impossible requirement in some WSNs.

In this paper, a solution based on the use of node behavior is presented. The cognitive features merged with WSN ones offer the possibility of collecting large amounts of information from the spectrum to model the behavior of each node. The spatial and temporal data redundancy makes it possible to use algorithms to detect changes in the behavior of each node in an unsupervised way.

4. Behavior-Based Systems

Like geolocation countermeasures, defenses based on behavior try to model the PU. The model is used to look for differences between a PU and attackers. For example, in [3] authors use some radio parameters to decide if the transmitter is an incumbent transmitter or an attacker. These parameters are as follows: transmitted power and location. For a typical TV scenario on CR the PU model can be very precise. However, as with geolocation countermeasures, the previous studies do not work with CWSNs. Unfortunately, a model for PU on CWSNs does not exist yet. PUs are usually more unpredictable than in previous scenarios. Moreover, the PU's behavior can be very different depending on the application. However, if we focus our CWSN on limited scenarios, for example, ambient intelligence in a home or a building, the PU is specifically defined. Parameters like power transmission, time occupancy of spectrum, and transmission frequency could be modeled.

Learned behaviors of these parameters allow the system to create some profiles which are compared with periodically acquired measures. It is easy to understand that, when a PUE attack happens, an anomaly in learned parameters can be detected. The intrinsic goals of an attacker make it impossible to have a complete likeness between a PU and a PUE attack. For example, if the goal of a PUE attack is the use of a whole frequency band, it needs to transmit more frequently, with more power and different types of packets than a normal PU.

In [10], the authors use the packet traffic to model the sensor behavior. The packet train size, packet train length, interpacket times, and payload size are used to characterize the packet traffic. They apply these profiles to detect anomalies, such as sinkhole attacks.

In [11], another approximation is taken to monitor the node's behavior. In this work, a group of capable nodes form the attack detection system (ADS) which analyzes the transmitted packets among its neighbors. The reason to limit the ADS to some nodes only is that a continuously monitoring node consumes much more energy than a normal one. Following the same idea, in [12], some monitor nodes sniff the communications in order to detect anomalies. They base their decisions on some principles of WSN such as message symmetry or node similarity.

Finally, in [13] the nodes create neighbor profiles according to the sequence of received packets. The attack is detected using the distance between sequences. The distance is calculated as the number of differences between them.

As a conclusion, the previous works use traffic monitoring to train a behavior model of the network. In this work, we can use other parameters such as power transmission to detect anomalies in CWSN. This is possible thanks to some cognitive features such as spectrum sensing and learning. The advantage of these parameters is that they can be used in more flexible networks or independently of the application. Another advantage over the previous works is the collaboration between nodes. The final decision in the detection of anomalies is collaborative. The more the nodes collaborating in the decisions, the better the PUE detection results. One important reason in order to use collaboration in this scenario is the ignorance of the attacker's position. If the system only uses the information of one or a few nodes, the node profiles might be wrong because of the attenuation or the distance between the SU and the attacker. For example, if an SU is still far from the attacker, it might not receive all the transmitted packets by the attacker. Moreover, the power received could be very variable because of the attenuation. The redundant information, inherent in WSN, and the collaboration in CWSN reduce the possibilities of errors in the sensed information, creating better profiles, and in the final decision. Another motivation for the collaboration is the resource limitation. Nodes of CWSN have to sleep, and their computing resources and energy are limited. During the sleep state, the nodes do not capture information. In these moments, other active nodes can capture information, and the profiles are developed with data from every time.

5. Assumptions and CWSN Scenario

Security is a rarely studied field in cognitive networks, and it is even less studied in CWSNs. But this does not mean that security is not important. On the contrary, security is important in WSNs and so will be in future cognitive applications such as health, home security, or military scenarios. Spectrum sensing is crucial in order to detect malicious behaviors in the transmissions or to analyze suspicious changes in the radio spectrum. The ability to learn and collaboration are also essential for many security algorithms. Finally, adaptation is the base of some countermeasures against jamming or routing attacks.

In our model, a CWSN consists of a set $S = \{s_1, s_2, \ldots, s_n\}$ of n cognitive wireless sensor nodes with different roles. Each node can communicate with other nodes within a certain range. In a common CR application, the PUs are usually a TV tower or a base station. In most cases, the SUs know the location and the transmission parameters of PUs, but with CWSNs we cannot assume that. The location and the radio parameters of the nodes are unknown. However, we assume that the nodes have a stationary behavior that allows them to learn from spectrum sensing.

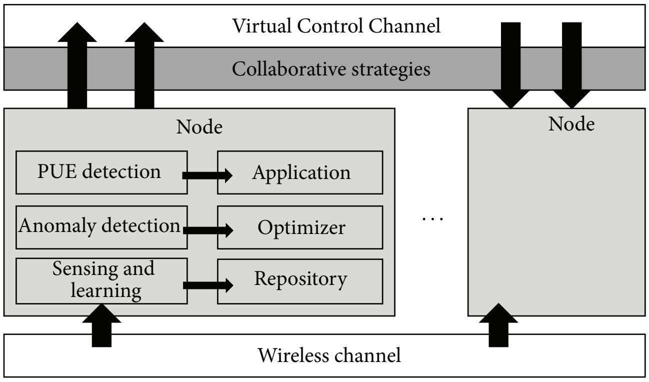


Figure 1: Cognitive features and modules responsible for them.

SUs and PUs act in CWSNs in different ways. While PUs take preference because they are responsible for critical sensors and information, SUs only send the information when the channel is empty or it satisfies some conditions. In a typical CWSN the number of nodes can usually vary between 5 and 200. For our study, we assume that networks with more than 200 nodes are not common.

In our scenario, spectrum sensing is carried out by multiple wireless modules that all nodes in the network have. More specifically, these interfaces work in the ISM bands (2.4 GHz and 868 MHz). All of them can extract information from the environment: received signal power, noise power, or time between packets. The information is processed, stored, and shared according to the implemented algorithm. We further assume the existence of a Virtual Control Channel (VCC) to share this information, with no extra overhead over regular cognitive communications.

Apart from primary and secondary users which form the network, the attackers are the key in security scenarios. The PUE attack in cognitive networks usually belongs to one of these two categories.

(i) *Selfish PUE Attacks*. In this attack, an attacker's objective is to maximize its own spectrum usage.

(ii) *Malicious PUE Attacks*. The objective of this attack is to obstruct secondary user's access to the spectrum.

Our PUE model is captured by the following set of assumptions.

(1) A PUE node is a wireless node with k wireless configurations (where k is the number of wireless configurations at each node NW).

(2) A PUE attacker has similar hardware and radio characteristics to the rest of nodes.

(3) The network does not have any information about the position of the PUE attacker or its strategy.

(4) The PUE attacker and the PUs cannot have exactly the same radio behavior.

As we explain in Section 4 we assume that, regardless of the kind of PUE attack, the malicious node has to change its behavior. If the node continues with the same behavior from the creation of the network and it uses exactly the same radio parameters, attack detection is impossible using either learning behavior or any other method.

6. System Architecture

The system architecture presented in Figure 1 makes use of the collaboration in order to achieve the anomaly detection goal. Its main characteristics are the distributed learning and the collaboration in the final decisions.

6.1. Spectrum Sensing and Learning. Spectrum sensing is the first module of the entire chain in the system. All the nodes in the system sense the radio spectrum and analyze the data to create a precise enough profile of each node. The spectrum sensing in this system consists of the detection of the signal level in each channel. Each node is aware of the spectrum occupancy in its near range. Moreover, the nodes are able to detect all the valid packets over a reception power threshold. Despite the fact that the packets are usually sent to a specific node, the rest of the nodes in a sensing stage can capture the packets and extract information from them such as the source, the sink, and the time stamp.

Cognitive wireless nodes have some constraints that limit the system when a data base has to be created. For example, low computational resources and low available memory do not allow for the creation of complex detection algorithms or the storage of large data bases.

We propose the nonparametric Cumulative Sum (CUSUM) algorithm [14] for the detection of changes in some key spectrum sensing captured features. The CUSUM is an algorithm used in WSN in order to detect changes in the mean value of a stochastic process. The advantages of this algorithm in CWSN are the low computational requirements and the no assumption of any previous knowledge about the PUE attack. As it has been explained in Section 4, if the scenario is limited, usually the sensor nodes have a stationary behavior. Moreover, the attack happens at unknown time. These are the reasons why the CUSUM algorithm is applicable in this approach.

In this case, some key features, such as the received power, are necessary to model the node behavior. A good approximation is to save the key parameters that define the feature. In this work, the number of measures, the average, and the variance are stored in each node repository. The average "X_n" and the variance "S_n" are calculated using only the previous one's value and the current sample as shown in

$$\overline{X_n} = \frac{1}{n}\sum_{i=1}^{n} x_i = \overline{X_{n-1}\frac{n-1}{n}} + \frac{x_n}{n},$$

$$\overline{S_n^2} = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \left|X_n\right|^2 = \overline{E_n^2} - \left|X_n\right|^2, \qquad (1)$$

$$\overline{E_n^2} = \overline{E_{n-1}^2\frac{n-1}{n}} + \frac{x_n^2}{n},$$

where $\overline{E_n^2}$ is the average of the squared values. So, each node creates a table with the following data:

$$\left\{\text{Node ID}, n, \overline{X_n}, \overline{E_n^2}\ \overline{S_n^2}, \right\}, \qquad (2)$$

Throughout the learning stage the nodes update and refine these values which will be used as the base in the anomaly detection algorithm.

6.2. Anomaly Detection. When the system has captured enough packets, the node profiles are ready to compare themselves against the new samples. During this step, the optimizer applies the CUSUM algorithm, compares the current samples with the average in the profiles, $\overline{X_n}$, and sends anomaly warnings to the application. The comparison between the samples and the profile is calculated according to the Euclidean distance. If the distance is lower than a number of standard deviations, sample is considered as a normal value. However, if the sample is out of the allowed range, the optimizer sends the anomaly warning to the application level. In this way, the algorithm can be configured with high threshold values, with low false positive rate and slow detection or with low threshold values that imply more false positives but a faster detection.

The application layer is responsible for managing anomaly warnings. Above the application layer, the whole system can be applied for any anomaly detection. In this work, the application filters the warnings and only creates a PUE attack warning when the anomaly continues for a configurable time. If the anomaly behavior in a node exceeds that time, the application marks the node as a possible PUE attacker.

6.3. Collaboration. The previous chapters describe how the nodes in the network can collect information from the spectrum as a key feature in cognitive solutions. The stored information used by an isolated node could be useful for a particular optimization, but if the final goal of the network is a general optimization of a parameter, in this case the security and collaborative strategies are essential. Collaboration strategies are a common solution in other cognitive fields like spectrum sensing and also in security scenarios, such as PUE detection. The next section shows how the introduction of collaborative detection significantly improves results.

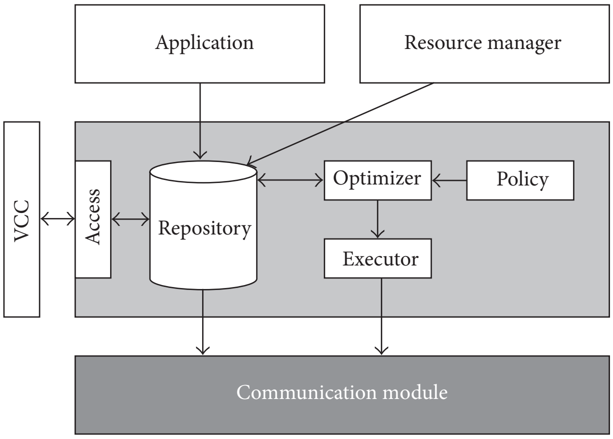


Figure 2: Cognitive radio module structure.

In this work, the SU nodes collaborate by sharing information about the detected anomalies. This information may be characteristic of spectrum sensing or anomalies detected by a single node. When an application marks a node as a possible PUE attacker, it sends a message through the Virtual Control Channel (VCC), a method for sharing information in cognitive networks. Finally, as we will explain later, the VCC allows other nodes to access to almost any information stored in other neighbor nodes.

7. Experimental Results

7.1. Simulation Tools. The proposed countermeasures have been tested on a CWSN simulator [15]. This simulator has been developed over the well-known Castalia simulator. The structure of Castalia has been improved to provide it with cognitive features. The CWSN simulator is responsible for the scenario definition, the simulation of the spectrum state, and the communication between nodes from the physical to the application layer. It supports the cognitive features in the cognitive module, shown in Figure 2, which has the following parts.

(i) *Repository*. It retrieves information about the local and/or remote nodes: information learned, decisions made, or current state. The kind of information stored depends on the context and the requirements of the system.

(ii) *Access*. This module lets a local repository access the repository of remote nodes. At the same time, it exports a subset of the local repository to remote nodes.

(iii) *Policy*. This enforces the requirements for the global system depending on several factors. In this paper, security is the policy to optimize.

(iv) *Optimizer*. This processes the repository information bearing in mind the requirements imposed by the policy module. Decisions regarding the behavior of the local node are the results of processing. They are stored in the repository and evaluated by the executor.

(v) *Executor*. This module performs the decisions made by the optimizer.

Furthermore, it provides the Virtual Control Channel (VCC), a new method for sharing cognitive information among the CR modules of the nodes. CR modules can access exported information from remote repositories through this channel. It allows CR modules to be aware of their surroundings and even of the whole network.

7.2. Simulation Experiments. The attacker is implemented as an SU that changes its behavior in a precise moment acting like a PU. The attacker will try to adapt all radio parameters according to the PU behavior. Some of them, such as modulation, encoding, or carrier frequency, probably will be exactly like those of the PU for two reasons. The attackers and the PUs usually have the same hardware characteristics; therefore the attackers can imitate the PU. The second reason is that the attackers do not need to change these parameters to reach their possible goals: to use more spectrum, to transmit information to other destinations, or to prevent SUs transmissions.

According to this, it is reasonable to restrict the parameters that the attackers will change to transmitted power and occupied spectrum bandwidth. In this work the received power has been used to detect anomalies, like a PUE attack in the network.

Setting this parameter to a similar value to those used by a real PU we can check how precise the algorithm is in detecting this kind of attacks.

In order to test the presented solution, when an attacker changes its behavior, the maximum allowed change in transmitted power is 1 dBm. Even with this small change, the system has demonstrated to be very efficient in detecting anomalies.

Several simulations have been executed in the simulator to extract results and to draw conclusions from the work. The scenarios have some common characteristics.

(i) The scenario area is a 30 m × 30 m square.

(ii) The complete simulation time is 500 seconds.

(iii) The number of nodes in the simulation varies between 50 and 200, including one server, 6 Pus, and a variable number of attackers.

(iv) The learning stage covers the first 60 seconds.

(v) The attacks start at second 100.

(vi) The SUs and PUs send information to the sink.

(vii) However SUs only send the information when the channel is not being used by any PU.

7.3. Results and Discussion. The first figures (Figures 3 and 4) present results about the learning speed of the systems. That means how much time the system needs to converge and to reach acceptable learned values.

The results of a network with 50 nodes are shown in Figures 3 and 4. Figure 3 represents how an SU learns about the power received from other node which has an abnormal operation. The node has a normal behavior during the first 30 transmitted packets. At this time, the attack starts, and the sensing power changes. As we can observe, the average is stable with a few samples. The top limit and the bottom limit form a range where the sensing data is considered normal. When the data is out of the limits, the node interprets it as an anomaly.

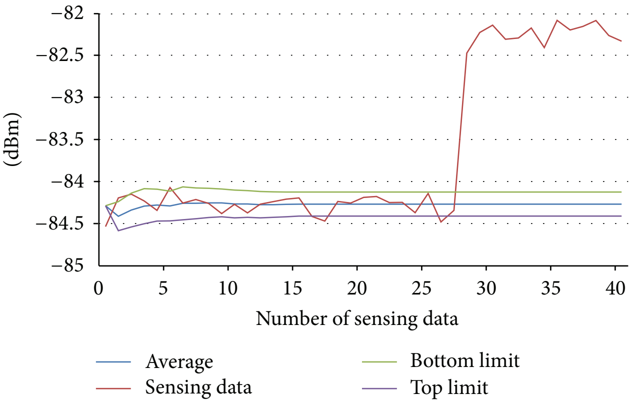


Figure 3: Sensing power and learning average from a PUE attacker node.

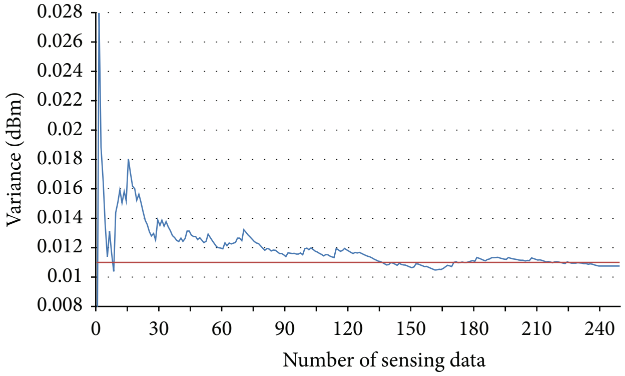


Figure 4: Learning power variance from PUE attacker node.

In the second figure, we can observe how fast the system learns. With few samples, the variance fluctuates but, when the node has more information, the variance stabilizes over 1%. The number of received packets that a node needs to refine the information is showed in the x-axis.

In the next figures (Figures 5, 6, 7, and 8), where the PUE attack scenarios are presented, the false positive parameter is presented. The system has shown very good behavior in detecting the attackers, with a detection rate over 98% in all simulations. However, for some combination of parameters, some normal nodes are detected as attackers. In Figure 5 we can see the results of a simulation with 50 nodes, including 5 PUE attackers, 6 PUs, 1 sink, and 38 SUs. In this situation the decisions taken individually by each node are complemented by the collaboration between nodes. Each line represents a different scenario whit the percentage of SUs that collaborate in the detection changes. The x-axis represents the number of standard deviations that a sensing power measure can deviate

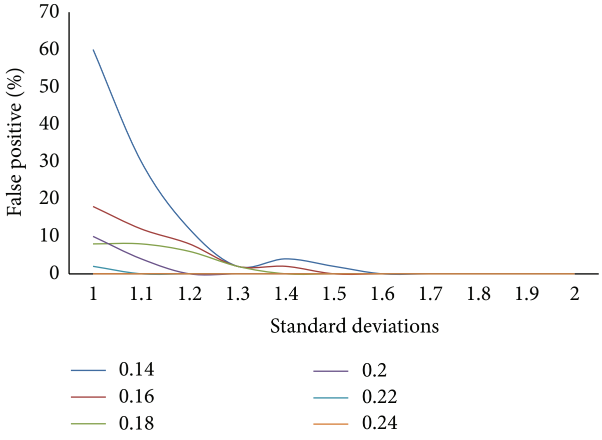


Figure 5: PUE detection results with 50 nodes.

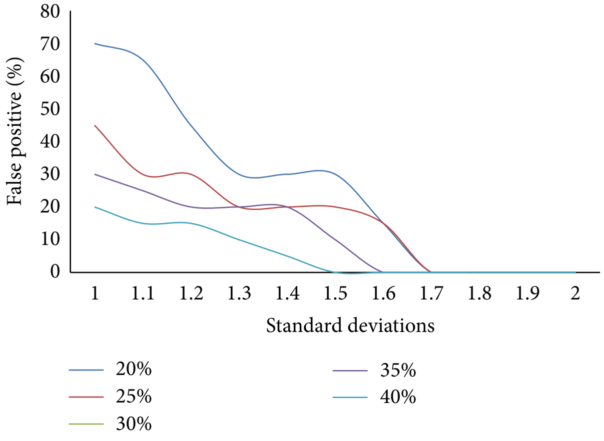


Figure 6: PUE detection results without filtering in the nodes.

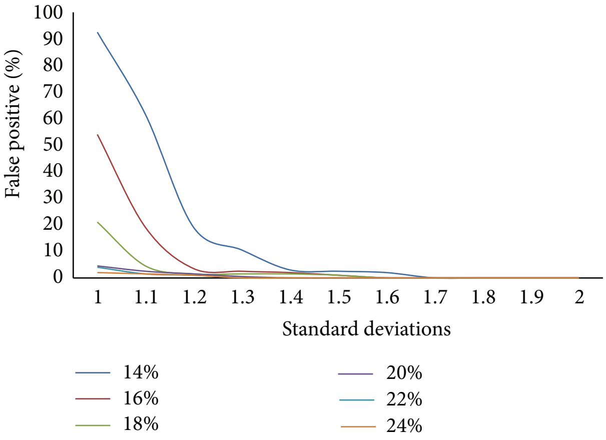


Figure 7: PUE detection results in a network with 200 nodes.

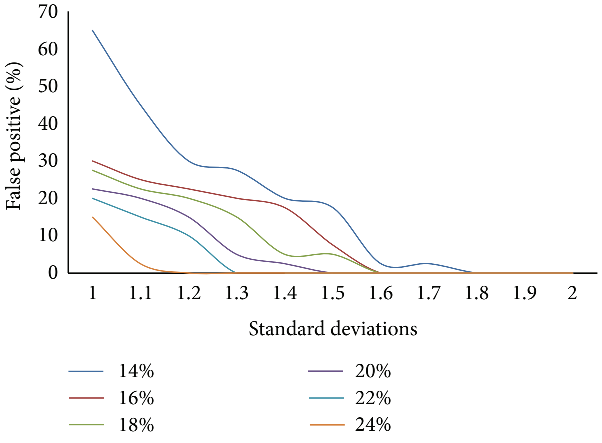


Figure 8: False positives in a multiple PUE attack.

from the learning average to be considered as a normal value. Finally, the y-axis represents the false positive percentage.

As we can appreciate, the percentage of collaborative nodes is essential in the PUE detection. For a percentage of around 20% of collaborating nodes the results are very good, with a false positive rate of under 10% with a margin of one standard deviation for anomaly warnings regarding the average in the profile. If we increase the parameter to 1.3, the results are very satisfactory with false positive and false negative rates near 0%.

Figure 6 shows another scenario with worse conditions than the previous one. In this case, the nodes send worse information than in the previous scenario to the other nodes. This is because the node's application does not filter the information received from the optimizer, as Section 6.2 explains, and sends too many anomaly warnings through the VCC. However, if the margin of standard deviations is increased to 1.5 and the number of collaborative nodes is over 30%, the results are good enough.

However, if the collaboration between nodes is eliminated and the filter in the nodes is improved, the system has shown poor results. The system is not capable of discriminating between the PUEs and normal behavior.

In order to prove the proper working of the system in larger networks, we have simulated a new scenario with 200 nodes. Figure 7 shows that, if the percentage of collaborating nodes is the same, the system keeps differentiating the PUE attack in almost every simulation, but the results become slightly worse. This is because more nodes in the same scenario space can produce more anomalies such as collisions, interference, higher noise level, or retransmissions.

As another interesting result, in Figure 8, the behavior of the system can be observed against a multiple PUE attack, where 10 malicious nodes attack the system after the learning time. In this case, where 25% of the nodes are attackers, the system behavior gets worse. But, even in this case, if the number of collaborative nodes is over 20%, the results are satisfactory.

The results conclude that the most important parameter to improve PUE detection is the number of collaborative nodes. Other parameters, such as the application algorithm or filter and the margin to mark data as anomalous, affect the results but to a lesser extent.

The same analysis has been studied using the bandwidth occupied by the nodes. In this case, the results are not good enough. The reason for the poor results is the behavior of the secondary users. As we have explained before, the secondary nodes only send packets when the channel is free, so the occupied bandwidth has a greater variance than in the power detection-based scenarios. The PUE attack has been impossible to detect with good precision using the occupied bandwidth. This only means that the presented algorithm does not work with our definition of the SUs.

8. Conclusions

In this paper, a new method of detecting PUE attacks on CWSNs has been described based on cognitive features such as sensing, learning, and collaboration. A new simulator has been used to develop the scenarios that prove that collaboration is essential for good anomaly detection. The results have been extracted and presented in the graphics shown in Section 7.3.

Different layers of cognitive architecture implement the tasks to achieve the final objective, PUE detection. Cognitive nodes sense the spectrum and create neighbor profiles in order to model the behavior. The information stored in the repository is used to warn the application about anomalous data. The application is responsible for filtering the information and collaborating with other nodes.

If the collaborative nodes are over 20% of the total, the PUE attack detection has satisfactory results, with a 98% of attacks detected and a false negative rate near 0%, independently of the number of nodes in the scenario.

As the results show, the collaborative systems and the behavior models are valid to detect a PUE attack when there are few PUE attacker nodes compared with the total number of nodes in the network, and we assume that PUE attacker should change its behavior in order to reach the malicious goals.

References

[1] J. Huang, G. Xing, G. Zhou, and R. Zhou, "Beyond co-existence: exploiting WiFi white space for Zigbee performance assurance," in *Proceedings of the 18th IEEE International Conference on Network Protocols (ICNP '10)*, pp. 305–314, October 2010.

[2] R. Chen and J.-M. Park, "Ensuring trustworthy spectrum sensing in cognitive radio networks," in *Proceedings of the 1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks (SDR '06)*, pp. 110–119, September 2006.

[3] R. Chen, J.-M. Park, and J. H. Reed, "Defense against primary user emulation attacks in Cognitive Radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 25–37, 2008.

[4] Z. Chen, T. Cooklev, C. Chen, and C. Pomalaza-Ráez, "Modeling primary user emulation attacks and defenses in cognitive radio networks," in *Proceedings of the IEEE 28th International Performance Computing and Communications Conference (IPCCC '09)*, pp. 208–215, December 2009.

[5] Z. Jin, S. Anand, and K. P. Subbalakshmi, "Detecting primary user emulation attacks in dynamic spectrum access networks," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–5, June 2009.

[6] Z. Yuan, D. Niyato, H. Li, and Z. Han, "Defense against primary user emulation attacks using belief propagation of location information in cognitive radio networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '11)*, pp. 599–604, March 2011.

[7] L. Huang, L. Xie, H. Yu, W. Wang, and Y. Yao, "Anti-PUE attack based on joint position verification in cognitive radio networks," in *Proceedings of the International Conference on Communications and Mobile Computing (CMC '10)*, pp. 169–173, April 2010.

[8] Z. Caidan, W. Wumei, H. Lianfen, and Y. Yan, "Anti-PUE attack base on the transmitter fingerprint identification in cognitive radio," in *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '09)*, pp. 1–5, September 2009.

[9] D. Hao and K. Sakurai, "A differential game approach to mitigating primary user emulation attacks in cognitive radio networks," in *Proceedings of the IEEE 26th International Conference on Advanced Information Networking and Applications (AINA '12)*, pp. 495–502, March 2012.

[10] Q. Wang, "Packet traffic: a good data source for wireless sensor network modeling and anomaly detection," *IEEE Network*, vol. 25, no. 3, pp. 15–21, 2011.

[11] L. Yang and H. Kai, "Behavior-based attack detection and reporting in wireless sensor networks," in *Proceedings of the 3rd International Symposium on Electronic Commerce and Security (ISECS '10)*, pp. 209–212, July 2010.

[12] F. Wang and J. Gao, "Behavior monitoring framework in large-scale wireless sensor networks," in *Proceedings of the IEEE 29th International Performance Computing and Communications Conference (IPCCC '10)*, pp. 138–145, December 2010.

[13] Q. Wang and T. Zhang, "Detecting anomaly node behavior in wireless sensor networks," in *Proceedings of the 21st International Conference on Advanced Information Networking and ApplicationsWorkshops/Symposia (AINAW '07)*, pp. 451–456, May 2007.

[14] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Anomaly detection in wireless sensor networks," *IEEE Wireless Communications*, vol. 15, no. 4, pp. 34–40, 2008.

[15] J. Blesa, E. Romero, J. C. Vallejo, D. Villanueva, and A. Araujo, "A cognitive simulator for wireless sensor networks," in *Proceedings of the 5th Internation Symposium of Ubiquitous Computing and Ambient Intelligence (UCAMI '11)*, December 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 834547, 8 pages
http://dx.doi.org/10.1155/2013/834547

Research Article

Evaluation, Energy Optimization, and Spectrum Analysis of an Artificial Noise Technique to Improve CWSN Security

Javier Blesa, Alvaro Araujo, Elena Romero, and Octavio Nieto-Taladriz

Electronic Engineering Department, ETSI Telecomunicación, Universidad Politécnica de Madrid, 28040 Madrid, Spain

Correspondence should be addressed to Javier Blesa; jblesa@die.upm.es

Received 1 March 2013; Accepted 4 June 2013

Academic Editor: S. Khan

This paper presents the security evaluation, energy consumption optimization, and spectrum scarcity analysis of artificial noise techniques to increase physical-layer security in Cognitive Wireless Sensor Networks (CWSNs). These techniques introduce noise into the spectrum in order to hide real information. Nevertheless, they directly affect two important parameters in Cognitive Wireless Sensor Networks (CWSNs), energy consumption and spectrum utilization. Both are affected because the number of packets transmitted by the network and the active period of the nodes increase. Security evaluation demonstrates that these techniques are effective against eavesdropper attacks, but also optimization allows for the implementation of these approaches in low-resource networks such as Cognitive Wireless Sensor Networks. In this work, the scenario is formally modeled and the optimization according to the simulation results and the impact analysis over the frequency spectrum are presented.

1. Introduction

One of the fastest growing sectors in recent years has undoubtedly been that of Wireless Sensor Networks (WSNs). WSNs consist of spatially distributed autonomous sensors that monitor a wide range of ambient conditions and cooperate to share data across the network. WSNs are increasingly being introduced into our daily lives. Potential fields of applications can be found, ranging from the military to home control commercially or industrially, to name a few. The emergence of new wireless technologies such as Zigbee and IEEE 802.15.4 has allowed for the development of interoperability among commercial products, which is important for ensuring scalability and low cost. Most WSN solutions operate on unlicensed frequency bands. In general, they use Industrial, Scientific, and Medical (ISM) bands, like the worldwide available 2.4 GHz band. This band is also used by a large number of popular wireless applications, for example, those that work over Wi-Fi or Bluetooth. For this reason, unlicensed spectrum bands are becoming overcrowded. As a result, coexistence issues on unlicensed bands have been the subject of extensive research, and in particular, it has been shown that IEEE 802.11 networks can significantly degrade the performance of Zigbee/802.15.4 networks when operating on overlapping frequency bands [1]. The increasing demand for wireless communication presents a challenge to make efficient use of the spectrum. To address this challenge, Cognitive Radio (CR) [2] has emerged as the key technology, which enables opportunistic access to the spectrum. A CR is an intelligent wireless communication system that is aware of its surrounding environment and adapts its internal parameters to achieve reliable and efficient communication. These new networks have a lot of applications, such as the cognitive use of the TV white space spectrum or making secure calls in emergency situations. Adding cognition to the existing WSN infrastructure brings about many benefits. However, cognitive technology will not only provide access to new spectrum bands but will also provide better propagation characteristics. By adaptively changing system parameters like modulation schemes, transmit power, carrier frequency, and constellation size, a wide variety of data rates can be achieved. This will certainly improve power consumption, network life, and reliability in a WSN.

The broadcast characteristic of the wireless medium makes it difficult to shield transmitted signals from unintended recipients. Security in wireless data transmission has traditionally been performed using cryptographic techniques at the network layer. The main drawback of this approach when deployed to WSN consists in their limited resources, which cannot support the execution of complicated encryption algorithms, resulting in shorter keys that are easier to discover. WSN nodes can also be captured and by using reverse engineering, attackers can turn them for use in their own benefit.

Physical-layer security has become a very interesting approach in the past few years [3]. The main idea behind physical-layer security is to limit the amount of information that can be extracted at the bit level by unauthorized receivers with the exploitation of all available channel state information (CSI). The fundamental problem of WSN is the difficulty to obtain a full CSI. The cognitive paradigm allows for spectrum monitoring and provides this information to the network. Cognitive features such as spectrum sensing, adaptation, and collaboration involve more resources in terms of energy consumption and spectrum saturation. Security mechanisms also make these problems worse. Therefore, the analysis, but also the optimization of the technique, is essential for Cognitive Wireless Sensor Network scenarios, where computational and energy resources are very limited.

Another difference between Cognitive Networks and CWSNs is the role of Primary Users (PUs) and Secondary Users (SUs). We assume a different behavior for the SUs and PUs in CWSNs because of the nature of these networks. For example, CWSNs usually operate on ISM bands, where anyone can transmit without a license. Because of this feature, the definition of Primary Users (PUs) and Secondary Users (SUs) should be different. For this CWSNs definition, the differences between PUs and SUs are based on the priority of their functionality. For example, a fire sensor would have more priority than a temperature sensor. While PUs take preference because they are responsible for critical sensors and information; SUs only send the information when the channel is free.

This paper presents the evaluation, the energy optimization, and the spectrum scarcity analysis of artificial noise techniques to increase physical-layer security in Cognitive Wireless Sensor Networks (CWSNs). These approaches introduce noise in the spectrum in order to hide the real information. They can operate independently of the higher layers in order to complement security. The limitation of resources is one of the motivations in order to integrate this security approach that complements higher-level techniques, such as encryption or specific routing. In other scenarios with fewer limitations, encryption could be a good solution, but in WSNs the encryption algorithms are not strong enough and advanced attackers can decrypt the real information easily. The introduction of noise into the area makes the obtaining and decryption process much more difficult for attackers.

The organization of this paper is as follows. Section 2 shows a related work about physical-layer security strategies. Section 3 indicates the specific characteristics of the presented scenario in the work. Then, Section 4 presents in more detail the approach and Section 5 the evaluation, the optimization, and the simulation results. Finally, Section 6 summarizes the main conclusions of this work.

2. Physical-Layer Security Approaches

The physical-layer security in CWSN is a new area where the investigation is in a very early state. Moreover, the energy optimization of these techniques has not been investigated yet. There are some previous researches about efficient transmissions [4] or spectrum sensing [5], but there is not energy optimization or spectrum analysis for security approaches in CWSN.

Therefore, in this section, we introduce schemes that could be used to achieve physical-layer security against different attacks in WSNs, which share common features with CWSN.

In recent years, the main issues of secure channel capacity have drawn much attention within the information theory community. Most of the works are focused on schemes to obtain the secrecy capacity with different Channel State Information (CSI) approaches. Barros and Rodrigues [6] developed a secure communication protocol to ensure wireless information-theoretic security based on common randomness via opportunistic transmission, message reconciliation, common key generation via privacy amplification, and, finally, message protection with a secret key. It was shown that the protocol is effective in secure key renewal even in the presence of imperfect CSI.

Other methods have been proposed to avoid attacks based on the exploitation of channel characteristics. The RF fingerprinting system implemented by Sperandio and Flikkema [7] consists of a multiple sensor system that captures and extracts RF features from each receiver signal. An intrusion detector processes the feature sets and generates a dynamic fingerprint for each internal source identifier derived from a few packets. This system monitors the temporal evolution and alerts when a strange fingerprint is detected. In [8] Li and Ratazzi propose a precoding scheme, in which the transmitted code vectors are generated by singular value decomposition of the correlation matrix, which describes the channel characteristic features between the transmitter and the intended receiver. Due to the differences in the multipath structure of the transmitter-receiver channels, even intruders with a perfect knowledge of the transmission code vectors are not able to acquire the true messages due to the difference between the locations of the intruders and the legitimate users.

Code approaches improve resilience against jamming and eavesdropping. In [9], a combination of turbo coding and the Advanced Encryption Standard (AES) cryptosystem is proposed. An error in the received ciphertext can cause a large number of errors in plaintext after decoding. Depending on the channel condition, this method can be adopted to choose the number of redundant bits required to protect the information, in order to achieve high efficiency. Another technique is the Spread Spectrum Coding, in which signal is spread by a pseudo-noise sequence over a wide-frequency bandwidth much wider than that contained in the frequency ambit of the original information. The main difference between convention cryptographic systems and spread-spectrum systems lies in their key sizes. Traditional systems can have a very large key space. However, in a spread-spectrum system, the key space is limited by the range of carrier frequencies and the number of different sequences. In [10], a method is proposed to enhance the physical-layer security of a CDMA system by using AES operations to generate the scrambling sequences. Data protection can also be provided using power approaches. The method proposed in [11] ensures secure communications depending on the channels condition. This method shows that a good secrecy can be achieved when the intruders channel is noisier than the receiver's channel. Artificial noise is generated using multiple antennas or the coordination of helping nodes and is injected into the null subspace of the intended receivers channel.

Discriminatory channel estimation is performed by injecting artificial noise to the left null space of the legitimate receivers channel to degrade the estimation performance of the eavesdropper [12]. By exploiting the channel feedback information from the legitimate receiver at the beginning of each communication stage, a multistage training-based channel estimation scheme is proposed [13]. It focuses on minimizing the normalized mean squared error of the channel estimation at the legitimate receiver subject to a constraint on the estimation performance attainable by the nonlegitimate receiver.

Most of these approaches can be improved using cognitive capabilities. The cognitive paradigm [14] provides a new scenario thanks to spectrum sensing, protocols to share information, and collaboration to optimize the communications.

The works presented in this section support the idea of artificial noise as a possible security option both in WSN and CWSN, but energy optimization and spectrum saturation are two fundamental keys in CWSNs that should be analyzed.

3. Assumptions and CWSN Scenario

In this section, the network and adversary models studied in this paper are described. This is necessary in order to define where the proposed technique is applicable.

3.1. Network Model. In our model, a CWSN consists of a set $S = \{s_1, s_2, \ldots, s_n\}$ of n cognitive wireless sensor nodes. Each node can communicate with other nodes within certain range.

All nodes have cognitive capabilities. Thus, each node has N_W different wireless configurations (physical channels and modulations). All sensors run some discovery routine and they can record the current spectrum state. We also assume that all these nodes can collude among themselves, sharing information and selecting the best available communication configuration. We further assume the existence of a Virtual Control Channel (VCC) to share this information, with no extra overhead over regular cognitive communications.

We consider a four-terminal system composed of a legitimate source (S) that sends the message to a legitimate destination (D), one or more relay nodes (R), and one or more eavesdroppers (E). Because of the cognitive capabilities there are no multipath routing schemes; only one path is chosen each time for the communication.

The above assumptions are light and realistic, considering usual WSN scenarios such us house monitoring, military, and infrastructure protection.

Table 1: Power consumption with different measurement methods.

	TX/RX mode (0 dBm)	Sleep mode
An ammeter	28 mA	160 uA
Dig. ammeter	27,7 mA	146,6 uA
Oscilloscope	26,7 mA	0 mA

3.2. Adversary Model. In this system model the adversaries are eavesdroppers. We assume a global adversary controlling some eavesdroppers which can collaboratively overhear all messages on the incoming and outgoing channels of the closer sensor nodes. Our eavesdropper model is captured by the following set of assumptions for the eavesdropper E: (i) E is a wireless node with N_K wireless configurations (where $N_K \leq N_W$); (ii) E can perform sophisticated spectrum processing with its available elements; (iii) network does not have any information about the position of E or its strategy; (iv) the eavesdroppers communicate among themselves via a separate channel invisible to the rest of the network. We point out that the assumption of a global eavesdropper who can monitor the entire network traffic is realistic for some applications.

We also assume that the eavesdroppers are passive; hence they cannot transmit noise signals. One eavesdropper can sense the spectrum using its different wireless configuration. They can collude among themselves, sharing information to obtain a global spectrum state.

We define the behavior of the eavesdroppers according to these assumptions:

(i) the eavesdroppers have cognitive capabilities, such as multiple interfaces or spectrum sensing,

(ii) the eavesdroppers sense the available channels and interfaces during a period of time. If they do not locate any signal they commute to the next channel or interface,

(iii) if an eavesdropper detects a signal, it will listen during a period of time in the same channel.

3.3. Power Consumption Model. The power consumption model used in this work represents the Texas Instruments CC2530 System-on-Chip Solution for 2.4 GHz IEEE 802.15.4 and ZigBee Applications transceiver [15] and the rest of the hardware consumptions. The hardware power consumption, excluding the radio interfaces, is modeled constantly because we assume it is negligible against the wireless transceivers ones.

The model has been completed with real data extracted from measurements in the Texas Instruments SmartRF05 evaluation board. These measurements include power consumption in transmission, reception, and sleep modes. Moreover, the model includes the power consumption in the transition between modes and the time spent on it.

The measurements in Table 1 are consistent with the theoretical values supported by the company. This table presents the measures captured in real devices that complement the simulator. The theoretical transmission power in 0 dBm is between 28.1 and 29.2 mA. The sleep mode consumption

Table 2: Delay transition matrix in CC2530.

	TX	RX	Sleep
TX	—	192 ms	120 ms
RX	192 ms	—	120 ms
Sleep	120 ms	120 ms	—

measure was the mode 2. Theoretically, the consumption in this mode is 190 uA. As we can see the transmission and reception modes consume the same energy, indicating the activation of the RF module.

Switch times among modes are based on the hardware specification. The delay transition matrix can be observed in Table 2.

4. Cooperative Artificial Noise Scheme

The key idea in this paper is that a transmitter, in cooperation with helper nodes, can artificially generate noise to conceal the secret message that it is transmitting. The noise is generated in such a way that only the eavesdropper is affected but the intended receiver is not because noise is generated in an orthogonal channel or modulation. This technique creates a difficulty in obtaining real information and also in the decryption process in the worst case, when the eavesdropper senses real packets.

CWSNs avoid one of the main constraints when using artificial noise injection techniques, the knowledge of the channel state information (CSI). In this approach, otherwise inactive nodes in the relay network can be used as cooperative artificial noise sources to confuse the eavesdropper and provide better performance in terms of security. In addition, we allow noncolluding eavesdroppers (E) to individually overhear the communication between S and D without any central processing. Based on [16] idea, adapting these concepts to new scenarios, a formal modeling is presented. Zhou and McKay present the base of this modeling in a multiantenna scenario. In this work, the scenario is formally modeled based on this mathematical base and adapted to a cognitive scenario, where the collaboration and the spectrum sensing can be used to improve the technique. Moreover, in [16] there are no relay nodes; the nodes can transmit at the same time for more than one antenna and there is no any presented optimization according the spectrum saturation or the energy consumption.

We denote the possible wireless configurations between S and D and between S and E as h and g, respectively, both of which are $1 \times N_W$ vectors, where N_W is the number of different wireless configurations in S:

$$\begin{gathered} h = \left\{h_1, \dots h_{N_W}\right\}, \\ g = \left\{g_1, \dots g_{N_W}\right\}. \end{gathered} \tag{1}$$

The elements of h and g are independent and identically distributed complex Gaussian random variables:

$$h_i, g_i = h_i(t), g_i(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-t/2\sigma}. \tag{2}$$

Knowledge of h is obtained using spectrum sensing capabilities. We assume that the knowledge of h and g is available at E, which makes the secrecy of the communication independent of the wireless configuration and the channel. S utilizes multiple wireless configurations to transmit the information-bearing signal into the receiver's channel, while simultaneously generating a noise-like signal into the null space of the receiver's channel. We let an $N_W \times N_W$ matrix $W = [w_1 \ W_2]$ be an orthonormal basis of $\mathbb{C}^{N_W}$, where $w_1 = h/\|h\|$ and W_2, is orthonormal to h. The transmitted symbol vector at S is given by $x = w_1 u + W_2 v$, where the variance of the information symbol u is σ_u^2 and the $N_W - 1$ elements of v are independent and identically distributed complex Gaussian random variables, each with a variance of σ_v^2. The information-bearing signal is represented by u and v represents the artificial noise. Therefore, the received symbols at D and E are given, respectively, by

$$y_D = hx + n = hw_1 u + hW_2 v + n = \|h\|^2 u + n, \tag{3}$$

$$y_E = Gx + e = Gw_1 u + GW_2 v + e, \tag{4}$$

where n and e are the additive white Gaussian noises (AWGN) in D and E with variances of σ_n^2 and σ_e^2, respectively. G is a matrix $N_E \times N_w$, where the N_E is the number of eavesdroppers. We see in (3) that w_1 spans the null space of h; hence the artificial noise v does not affect the received signal at D.

However, the received signal in E is a combination of two unknown Gaussian distributions. Even though E knows h and G channel affection, it does not know the weights W, so it is impossible for it to distinguish the signal u and the noise v. We consider a total power per transmission denoted by P:

$$P = \sigma_u^2 + (N_W - 1)\,\sigma_v^2. \tag{5}$$

We refer to P/σ_n^2 as the transmission signal-to-noise ratio (SNR). As an important characteristic of this scheme and a conclusion of the model, the artificial noise is always generated in the orthonormal channels of the information. Therefore, we assume that the noise does not affect the legitimate transmissions.

5. Results

In order to compare the security using this cooperative artificial noise technique with systems without physical-layer security strategies, metrics are necessary. For this purpose the secrecy rate and the secrecy outage probability are defined. The secrecy rate is the rate of transmission on the main channel that remains undecodable to the eavesdropper. When larger networks with multiple transmitters/receivers/eavesdroppers, as well as additional nodes such as relays, are considered, we can define the corresponding secrecy rate regions, or the aggregate secrecy sum rate.

A performance metric suitable for nonergodic channels is the secrecy outage probability (SOP), which describes the probability that a target secrecy rate is not achieved. The SOP characterizes the likelihood of simultaneously reliable and secure data transmission.

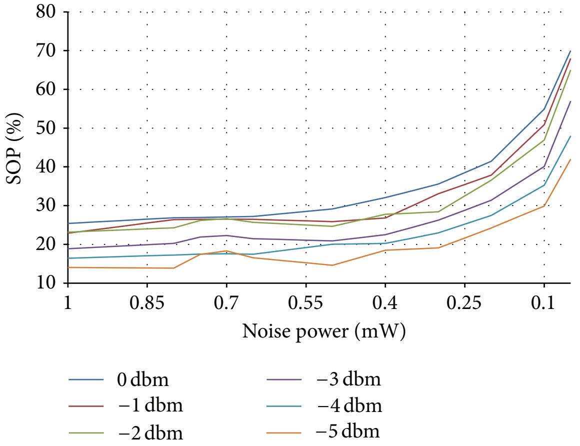


Figure 1: SOP for different emitter and noise power with 5 jamming nodes.

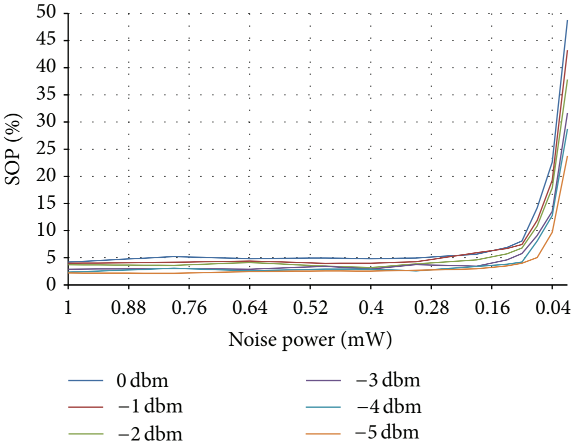


Figure 2: SOP for different emitting and noise power with 20 jamming nodes.

The efficacy of this scheme for different example scenarios using these metrics is presented. In order to simulate the attacks and the countermeasures, a CWSN simulator [17] has been used. This simulator has been developed over the well-known Castalia simulator [18]. Our modifications improve Castalia and include new cognitive features. The CWSN simulator responsibilities are scenario definition, simulation of the spectrum state, communications between nodes, and implementation of cognitive behaviors, attacks, and countermeasures.

Ten scenarios have been executed in the simulator to extract results and to draw conclusions from the work. The scenarios have some common characteristics.

(i) The scenario area is a 50 m × 50 m square.

(ii) The complete simulation time is 100 seconds.

(iii) The number of nodes in the simulation varies between 4 and 53, including one emitter node (S), one destination node (D), one eavesdropper attacker (E), and a variable number of secondary users that implement the relay and the jamming functions (R).

(iv) The emitter node sends 1 packet/s.

(v) The attack starts at the beginning of the simulation.

(vi) The eavesdropper nodes sense each channel for 50 ms. If in this time they do not detect a signal, they change to the next channel. However, if a signal is detected the eavesdropper will listen to the same channel for five seconds.

The variable parameters in the scenarios are as follows:

(i) the number of relay nodes (R),

(ii) the power transmission of the relay nodes (R),

(iii) the power transmission of the emitter node (S).

We have developed some graphics that summarize the results. In Figure 1, SOP for different emitter and noise power is presented. The number of nodes in the simulation is (7), including one source user, one destination node, and one eavesdropper in a 50 × 50 meters scenario. As we can see in the graph, there is a zone that provides the optimal relation between security SOP metric and noise power. Less noise power implies higher SOP rates. More noise power does not improve the security enough, and it affects the energy consumption and the spectrum performance.

In order to determine the influence of collaboration nodes over the network behavior, a new scenario has been simulated. Using the same 50 × 50 meter zone, the number of nodes in the simulation is 23, including one source user, emitting 1 packet/s, one destination, and one eavesdropper with 20 jamming nodes, emitting a rate between 1 and 1.1 packet/s. The shape in Figure 2 is similar to the one in the previous scenario. For high noise power the SOP level is lower than the one with 5 jamming nodes, between 2 and 5%, depending on the transmission power. This SOP is maintained for very low noise power, beyond 0.1 mW. There is also a zone where the relation between emission and noise power is optimal. This zone is shown in Figure 2.

Once the technique has been validated for security reasons and the results have been analyzed, the optimization and the impact in energy consumption should be analyzed. The next results show the energy optimization that consists of searching for the minimum resource utilization in order to achieve the security goals.

Figure 3 represents the additional power consumption that the jamming nodes introduce into the network. As we can see, there is almost no dependence between the emitter power and the power consumption of the jamming nodes. The quadratic form of the curve is derived from the omnidirectional antenna that simulates the nodes, with a circular range. The number of nodes affected by the transmissions of the others increases with r^2. In Figure 3 it can be appreciated that the network without using the noise technique has an energy consumption over 65 joules. This value can be increased to 250 Jules if the sensors that emit noise transmitted 1 mW. This is an additional energy consumption of more than 350%.

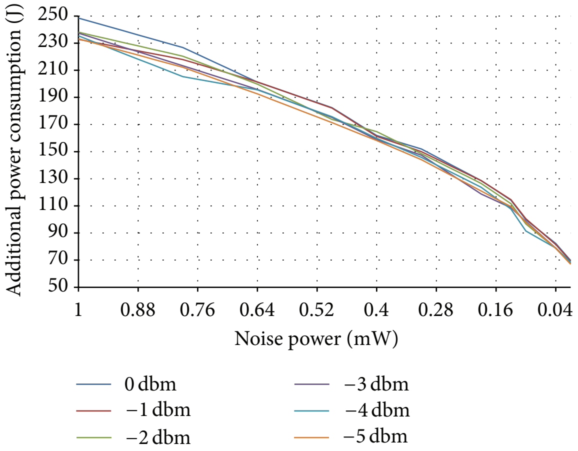


Figure 3: Additional power consumption in the network with 20 jamming nodes.

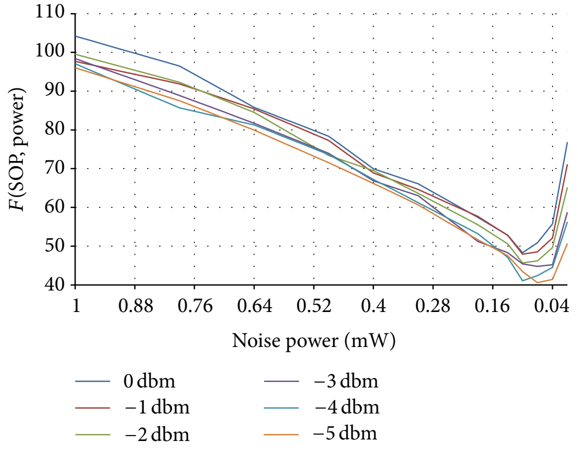


Figure 4: Jamming power variable. Function of SOP and additional power with $A = 1$ and $B = 1$.

If we can weigh the security and the consumption of the system, a formula similar to the next one will be a good solution:

$$F(\mathrm{SOP}, \mathrm{POWER}) = A \cdot \mathrm{SOP} + B \cdot \mathrm{POWER}, \qquad (6)$$

where POWER is the additional consumption ponderated to 100 like the SOP. A and B are the weights that the designer can control to give more importance to the security or the power consumption. Depending on these weights, Figure 4 has a different minimum. Figure 4 represents formula (6) with $A = 1$ and $B = 1$. Table 3 summarizes some optimum results for different values of A and B. As we can appreciate, using a power noise transmission of 0.07 mW, the SOP can be reduced to 4.5% in the best case.

A different approach is to modify the number of jamming nodes keeping the jamming power constant. Figure 5 shows the SOP for different simulations with a variable number of jamming nodes, from 1 to 19. As in the previous simulations, there is a zone where the SOP is very high, when the number of jamming nodes is low, and a zone where the SOP does not decrease a lot, with more than 5 nodes.

Table 3: Optimum values for different weights.

A	B	Pjamming	SOP	Power
1	3	0,01 mW	48,8%–23,76%	69 J
1	2	0,04 mW	22,6%–11,73%	77 J
1	1	0,07 mW	11%–4,5%	95 J
3	1	0,1 mW	8,1%–4%	100 J
10	1	0,19 mW	5,6%–3%	128 J

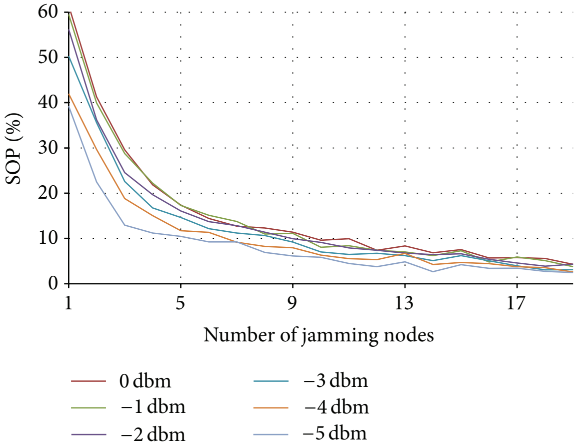


Figure 5: SOP for different number of jamming nodes.

The additional power consumption has a quadratic form, similar to that in Figure 3. It increases with the number of jamming nodes. If we create the same formula as in the first experiments, the results are shown in Figure 6.

A conclusion extracted from Figure 6 is that the use of more than 5 jamming nodes does not provide improvements in the optimizer function.

Figure 7 represents the curves for different values of A and B when we select a constant emitter power of 0 dBm. It can be seen that if the security (A) has more weight in the formula than the consumption (B), the minimum solution requires a higher number of jamming nodes. It can be observed that the optimal solutions for the weights represented in Figure 7 are between 3 and 7 jamming nodes. This indicates that increasing the number of jamming nodes makes the network consume more energy without benefits in security.

Finally, to complete the analysis with power consumption, we have simulated a new scenario in order to analyze the behavior of the system when spectrum saturation is higher. In this case, the simulation has 10 emitter nodes transmitting 10 packets/s. This is 100 times more traffic than in the previous simulations. We have observed that the results have a similar distribution with the difference that in this case the SOP decreases drastically. However, this is due to the attack model in which the listening time in a channel with signal is much longer than the sensing time for each channel. This is an important result because spectrum efficiency is the most important goal of cognitive radio. If this approach

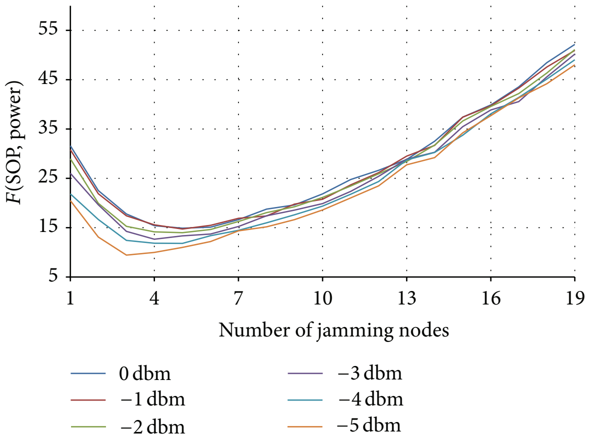


Figure 6: Number of jamming nodes variable. Function of SOP and additional power with $A = 1$ and $B = 1$.

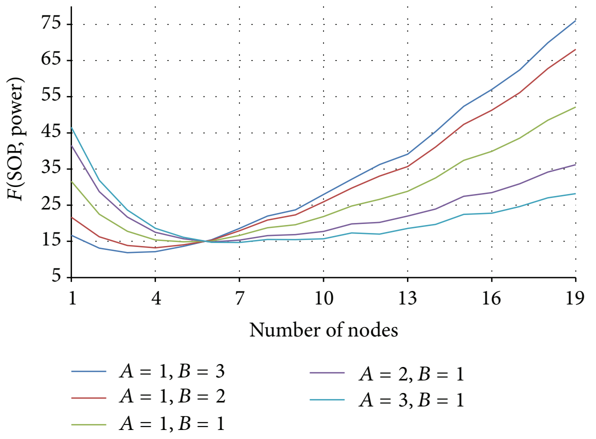


Figure 7: Number of jamming nodes variable and emitter power 0 dBm. Function of SOP and additional power with different values of A and B.

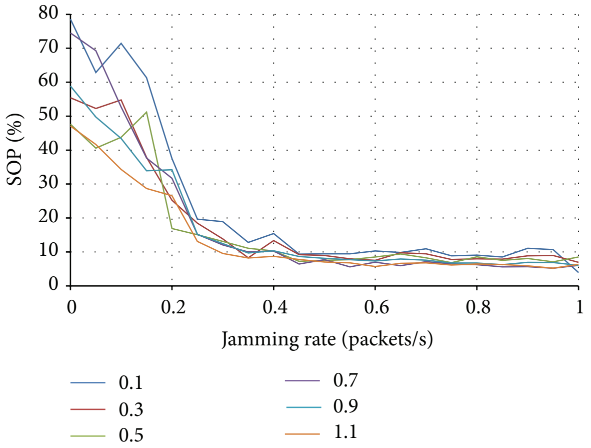


Figure 8: SOP for different emitter and jamming rates.

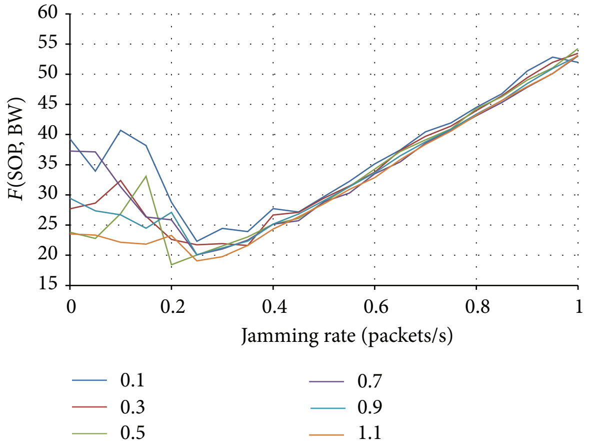


Figure 9: Function of SOP and jamming rate with $A = 1$ and $B = 1$.

significantly affects the spectrum occupancy, it would not be feasible.

A similar analysis has been made with spectrum utilization. This is an important resource that cognitive radio tries to use in a more efficient way. Therefore, this approach should be analyzed in order to have the minimum possible affect.

Figure 8 shows the SOP results for a variable jamming and emission rate. As we can appreciate, for low jamming rates (<0.2 packets/s) the SOP decreases quickly, but for higher values (>0.4 packets/s) the SOP is constant. This indicates that the optimum value in order to preserve the spectrum will be between 0.2 and 0.4 packets/s.

Figure 9 shows the conclusions specified before. Here, as in the power consumption experiments, an objective function is represented by (7). The figure shows that the optimal jamming rate for all the simulated scenarios is between 0.20 and 0.30 packets/s when $A = B = 1$. To increase the jamming rate above 0.30 packet/s costs more spectrum saturation than security benefits:

$$F(\text{SOP}, \text{POWER}) = A \cdot \text{SOP} + B \cdot \text{BW}. \quad (7)$$

Finally, Figure 10 and Table 4 show the optimum values for the objective function when the emission rate is constant and the weights A and B change. Here have been analyzed low packet rates, under 1 packet/s. The SOP rates are under 10% with a jamming rate over 0.45 packet/s.

6. Discussion and Conclusion

In this paper, we present an evaluation and energy consumption optimization and spectrum scarcity analysis of a cooperative artificial noise injection strategy for physical-layer security in multiuser cognitive wireless sensor networks as a supplement to encryption at higher layers. The generation of artificial noise makes the extraction of information from the spectrum difficult but also complements the upper layers security mechanisms, such as cryptography, making more difficult the decryption process of a noisy signal. According

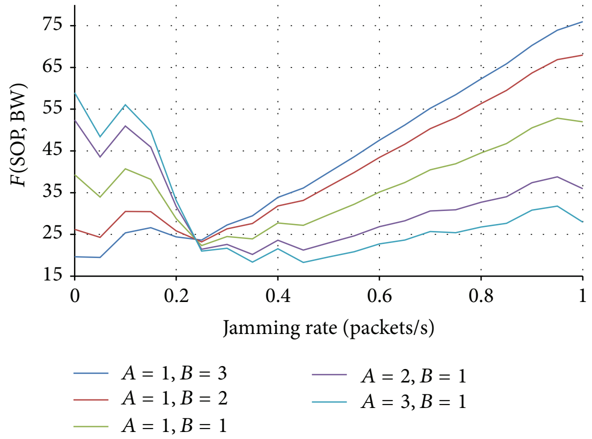


Figure 10: Function of SOP and jamming rate with different values of A and B and emitter rate 0.1 packets/s.

Table 4: Optimum values for different weights with jamming rate variable.

A	B	Jamming rate	SOP
1	3	0,05 pps	62,8–41,6%
1	2	0,25 pps	19,6–11,1%
1	1	0,25 pps	19,6–11,1%
2	1	0,35 pps	12,8–8,2%
3	1	0,45 pps	9,35–77,75%

to the CWSN scenario it is necessary to optimize the noise generation. Because of the CWSN nature noise affects power consumption and spectrum occupancy.

Cooperative artificial noise strategies with assistance from external helpers or inactive neighboring nodes are seen to be highly effective for increasing the secrecy of the transmitted data. Of course, a trade-off between energy consumption (additional sensor power consumption spent on transmission of noise), spectrum scarcity, and security level is necessary.

A cognitive simulation framework has been used to simulate the different scenarios. The eavesdropper attack model is always the same. The work is focused on the artificial noise strategy and energy optimization but in the future it will be interesting to hold a study with different attack models.

From the simulation results, we showed that there are different optimum solutions according to an objective function with different weights for energy consumption, spectrum scarcity, and security.

References

[1] J. Huang, G. Xing, G. Zhou, and R. Zhou, “Beyond co-existence: exploiting WiFi white space for Zigbee performance assurance,” in *Proceedings of the 18th IEEE International Conference on Network Protocols (ICNP ’10)*, pp. 305–314, October 2010.

[2] J. Mitola III and G. Q. Maguire Jr., “Cognitive radio: making software radios more personal,” *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[3] Y.-S. Shiu, S. Y. Chang, H.-C. Wu, S. C.-H. Huang, and H.-H. Chen, “Physical layer security in wireless networks: a tutorial,” *IEEE Wireless Communications*, vol. 18, no. 2, pp. 66–74, 2011.

[4] J. Mao, G. Xie, J. Gao, and Y. Liu, “Energy efficiency optimization for cognitive radio mimo broadcast channels,” *IEEE Communications Letters*, vol. 99, pp. 1–4, 2013.

[5] A. Sultan, “Sensing and transmit energy optimization for an energy harvesting cognitive radio,” *IEEE Wireless Communications Letters*, vol. 1, no. 5, pp. 500–503, 2012.

[6] J. Barros and M. R. D. Rodrigues, “Secrecy capacity of wireless channels,” in *Proceedings of IEEE International Symposium on Information Theory (ISIT ’06)*, pp. 356–360, July 2006.

[7] C. Sperandio and P. G. Flikkema, “Wireless physical-layer security via transmit precoding over dispersive channels: optimum linear eavesdropping,” in *Proceedings of Global Information GRID—Enabling Transformation Through 21st Century Communications (MILCOM ’02)*, vol. 2, pp. 1113–1117, October 2002.

[8] X. Li and E. P. Ratazzi, “Mimo transmissions with information-theoretic secrecy for secret-key agreement in wireless networks,” in *Proceedings of Military Communications Conference (MILCOM ’05)*, vol. 3, pp. 1353–1359, October 2005.

[9] Y. Hwang and H. C. Papadopoulos, “Physical-layer secrecy in AWGN via a class of chaotic DS/SS systems: analysis and design,” *IEEE Transactions on Signal Processing*, vol. 52, no. 9, pp. 2637–2649, 2004.

[10] G. Noubir, “On connectivity in ad hoc network under jamming using directional antennas and mobility,” in *Proceedings of International Conference on Wired /Wireless Internet Communications*, Lecture Notes in Computer Science, pp. 186–200, Springer, 2004.

[11] S. Goel and R. Negi, “Secret communication in presence of colluding eavesdroppers,” in *Proceedings of Military Communications Conference (MILCOM ’05)*, vol. 3, pp. 1501–1506, October 2005.

[12] T. H. Chang, Y. W. Hong, and C. Y. Chi, “Training signal design for discriminatory channel estimation,” in *Proceedings of Global Telecommunications Conference (GLOBE-COM ’09)*, pp. 1–6, IEEE, 2009.

[13] I. Csiszar and J. Korner, “Broadcast channels with confidential messages,” *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 339–348, 1978.

[14] S. Haykin, “Cognitive radio: brain-empowered wireless communications,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, 2005.

[15] “CC2530—2.4 GHz System-on-Chip for ZigBee, RF4CE, Smart Energy applications,” Texas Instruments, 2013, http://www.ti.com/ww/en/analog/cc2530/.

[16] X. Zhou and M. R. McKay, “Secure transmission with artificial noise over fading channels: achievable rate and optimal power allocation,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 8, pp. 3831–3842, 2010.

[17] J. Blesa, E. Romero, J. Vallejo, D. Villanueva, and A. Araujo, “A cognitive simulator for wireless sensor networks,” in *Ubiquitous Computing and Ambient Itelligence*, 2011.

[18] A. Rastegarnia and V. Solouk, “Performance evaluation of Castalia Wireless Sensor Network simulator,” in *Proceedings of the 34th International Conference on Telecommunications and Signal Processing (TSP ’11)*, pp. 111–115, August 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 397926, 8 pages
http://dx.doi.org/10.1155/2013/397926

Research Article

Impersonating-Resilient Dynamic Key Management for Large-Scale Wireless Sensor Networks

Gowun Jeong,[1] Yong-Ho Seo,[2] and Hyun S. Yang[1]

[1] *AI and Media Laboratory, Department of Computer Science, Korea Advanced Institute of Science and Technology, Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea*
[2] *Department of Intelligent Robot Engineering, Mokwon University, Doanbuk-ro, Seo-gu, Daejeon 302-729, Republic of Korea*

Correspondence should be addressed to Yong-Ho Seo; yhseo@mokwon.ac.kr

Received 18 March 2013; Accepted 2 June 2013

Academic Editor: S. Khan

Key management in a large portion of ubiquitous sensor networks has been a challenge due to the limited capabilities of their wireless communicating and battery-powered sensors. Moreover, an attacker physically capturing even a few nodes hampers the entire network security by impersonating nodes to inject false data in an undetected manner. To efficiently protect from such impersonating by node capture, we propose a new dynamic key management framework particularly for large-scale clustered sensor networks. In the framework, different keying mechanisms, respectively, secure in-cluster, intercluster, and individual communication by refreshing keys on demand, while adaptively handling node addition and capture. Theoretic analysis and simulation results show that our proposed framework provides higher connectivity and security against impersonating than other existing studies do, for better trade-off with resource overheads.

1. Introduction

Wireless sensor networks (WSNs) of wireless communicating and battery-powered sensors have attracted attention from ubiquitous networking due to such sensors' cheapness and handy installation. In particular, for unmanned monitoring, these networks are often deployed in unattended and adversarial environments [1]. Here rises providing security against security attacks, which are more likely to incur in such WSNs due to the sensors' limited capabilities on communication, computation, and storage, as a key issue. Among varied security attacks introduced in [2], we particularly target *impersonating by physical node capture* because such an attack enables attackers to compromise all the secrets, such as cryptographic keys, of captured nodes and spread malicious data out over the entire network with impersonating the captured nodes by the obtained keys. Thus, as several studies [2–5] have already noted, any security strategies to be proposed should be highly resource-efficient as well as provide the basic security requirements: *confidentiality*, protection of the content of a packet; *authentication*, corroboration of the source of a packet; and *integrity*, ensuring that the content of a packet is unchanged during transmission.

To achieve all of them, a lot of security schemes have been proposed based on symmetric or asymmetric cryptography. Simply speaking, the difference between them is if the same key is employed both by a sender for encryption and by its receiver for decryption or not [4, 6]. Although asymmetric schemes generally provide stronger authentication [2, 4], symmetric key algorithms have been superior in WSNs for their light complexity [5, 7, 8]. For instance, one of typical sensors Tmote has 10 kb RAM, 48 kb flash memory, 1 mb storage, and 250 kbps communication bandwidth, which is insufficient to enable traditional asymmetric cryptography to work [9].

In the paper, we propose a new symmetric key-based security framework for efficiently secure communication in WSNs. To suggest an efficient data aggregation model for large-scale WSNs, we first assume that a WSN consists of a single very powerful base station (BS) and a number of clusters of regular sensors as in [10, 11]. Accordingly, as in

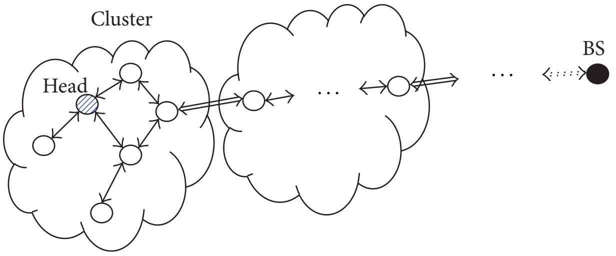


Figure 1: Three different communication patterns. Whereas the single-line arrows represent pairwise in-cluster communication, the double-line ones correspond to intercluster communication across two different clusters. The base station (BS) individually communicates with a node apart by starting with accessing to the closest node as the dotted-line arrow.

Figure 1, our framework supports three different communication patterns invoked by the following three types of keys: a static *individual key* shared with BS; a one-time *in-cluster key* shared within a cluster; and a static *intercluster key* between two neighbouring nodes across different clusters. We demonstrate how different keying mechanisms for these keys work to cope with impersonating by node capture in the following organisation. Section 2 describes what assumptions we take first and introduces our proposed security framework in cluster-based WSNs of our interest. In Section 3, with designing the impersonating attack model, we provide theoretic analysis of our framework in several performance metrics. Section 4 compares the performances of ours with those of selected conventional key schemes. Finally, we conclude the paper in Section 5.

2. Our Proposed Security Framework

Our framework provides multiple keying mechanisms to support the three communication patterns, while giving confidentiality, authentication, and integrity. We first give several assumptions regarding WSNs of our interest; present the keying schemes to establish and manage each of individual, in-cluster, and intercluster keys with secure transmission employing the keys; and discuss how to handle node addition and eviction.

2.1. Network and Security Assumptions. In the paper, a WSN consists of a single BS and a number of static wireless sensors that are deployed as in [12]. Every time a helicopter stays in a different deployment point, it scatters a sensor subset, called a *cluster*. In such a cluster, every node knows all of its cluster members' IDs and directly or indirectly interacts with others in a hop-by-hop manner. One of them is safely announced as the cluster head, a local controller, to the other members. When a node newly joins one cluster after the initial deployment, the head is informed of its ID by BS and lets all the members know the ID as well before it is actually placed. Since any sensor does not know its immediate neighbours in advance, it attempts to find its neighbours and establish required keys shortly after its deployment. This keying phase is assumed to be fairly well protected. However, once a security attacker captures a node, it can obtain all of the node's cryptographic information. We also assume that the most feasible path from a source to its destination is selected and notified to all the on-path nodes by the clusterbased backpressure routing algorithm of [13].

We illustrate our security framework using the next notations that appear in the rest of this discussion.

(i) N is the number of nodes in a WSN.

(ii) N_c is the number of clusters in a WSN.

(iii) D_c is the average size of a cluster.

(iv) D_n is the average number of a node's neighbours.

(v) D_b is the average number of *border nodes*, which relay messages to the outside of its cluster, in a cluster.

(vi) i and j are principals for clusters.

(vii) u and v are principals for sensor nodes.

(viii) $f_K(s)$ is a pseudorandom function (PRF) based on seed s with key K.

(ix) $h(K)$ is a one-way function (OWF) generating a one-way key from key K.

(x) K^u is the individual key of node u.

(xi) $\mathbf{C}^i = \{C_l^i\}$ for $0 \le l < k$ is a one-way key chain shared within cluster i.

(xii) b^u is the base of node u based on which u utilises its key chain in its own manner.

(xiii) t^u is a packet sequence number, which increases as time goes by, of node u.

(xiv) O^u is a one-time in-cluster key of node u.

(xv) K^{uv} is the intercluster keys of border node u with border node v of another cluster.

(xvi) $\{p\}_K$ is the encryption of packet p with key K.

(xvii) $[p]_K$ is the collision-free message authentication code (MAC) of packet p with key K. Given p and $[p]_K$ with known K, the receiver authenticates the sender and ensures p's integrity by confirming that the given p can generate the same MAC value as the arrived $[p]_K$.

(xviii) $d_1 \| d_2$ is the concatenation of data d_1 and d_2 by concatenator $\|$.

2.2. Individual Key. Since individual communication with BS contains highly private information, such as new node's joining announcement by BS or any neighbour's misbehaviour report by a node, which can be detected by any anomaly-based intrusion detection system discussed in [14], one's individual key is only shared with BS to authenticate the source of such notification. These keys are assigned by BS prior to deployment as follows.

Key Predistribution. To assign a unique individual key to each node, BS first builds a symmetric key matrix of $N_c \times N_c$, where every key pair of (i, j) and (j, i) such that $i \neq j$ is identical. Every (i, i) key is used to compute a distinct individual key for each member u of cluster i by $K^u = f_{(i,i)}(u)$. Before its deployment, every node is preloaded not only such an individual key but also $\{(i, j)\}$ for its cluster i and $1 \leq j \neq i \leq N_c$ to be utilised in the initial intercluster key establishment.

2.3. In-Cluster Key. We note that in-cluster communication most frequently occurs to exchange successive incoming percepts and to efficiently aggregate data. If a single cluster key is shared within a cluster for economical reasons, an adversary easily endangers the entire cluster by capturing only one node. Thus, we propose that a group of nodes sharing a key chain utilises all of the keys as each one's one-time encrypting keys in their own manners. One's current one-time key is derived only by its neighbours in its cluster, called *in-neighbours*, with its privately known base in advance and the current packet sequence number unless the base is compromised. This idea has, over the conventional key chain studies [1, 15, 16], the following additional advantages: strong key freshness, no need of key disclosure synchronisation, and no message overhead for direct key delivery.

Key Chain Predistribution. Before the initial deployment, BS generates and provides a unique one-way key chain of k keys for every cluster i based on OWF h and key (i, i) of the key matrix. As in μTELSA, $\mathbf{C}^i$ is constructed by $h((i, i)) = C^i_{k-1}$, $h(h((i, i))) = h^2((i, i)) = C^i_{k-2}, \ldots$, and $h^k((i, i)) = C^i_0$. To every member u, BS randomly assigns neighbour-distinct base b^u in $[0, k)$, based on which u has its own key use order as in Figure 2. Due to the hard-to-reverse nature of OWF, this generation-reversing key-use guarantees that any lower indexed key hardly implies higher indexed keys.

Neighbour Discovery and Base Exchange. After its deployment, every node u first attempts to find its any neighbour v by broadcasting its id, cluster ID i, and random key index r in $[0, k)$ in public as well as its base in private by key C^i_r as (1). Receiver v in the same cluster replies as (2). Otherwise, v in another cluster j does as (3)

$$u \longrightarrow * : u, i, r, \{b^u\}_{C^i_r}, \qquad (1)$$

$$v \longrightarrow u : v, \{b^v\}_{C^i_{r+1}}, \qquad (2)$$

$$v \longrightarrow u : v, j. \qquad (3)$$

Having received packets as (3), u becomes aware that it is a border node.

Rechaining and Key Chain Distribution. When member v reports its key reference exhaustion on the currently shared key chain to the head of cluster i or when any member v is perceived as captured, the head u generates key $K' = f_{K^u}(v)$ and new $\mathbf{C}^i$ by k recursions of $h(K')$. To propagate this key chain, u conveys K' after encrypting it with its current O^u of the old key chain as follows:

$$u \longrightarrow * : t^u, \{K'\}_{O^u}, [K']_{O^u}. \qquad (4)$$

Every receiver can generate the same key chain based on the arrived K' and passes it on to its in-neighbours in the same manner. As soon as a nonborder node forwards K' to others or a border node generates the new key chain, K' and the old key chain are immediately erased and every sequence number is reset to 0.

2.4. Intercluster Key. For packets crossing clusters, every pair of border nodes in two adjacent clusters, called *interneighbours*, should share a distinct pairwise key. So far, every node has been loaded $(N_c - 1)$ keys of the key matrix and known if or not it is a border node. Every nonborder node immediately erases the keys because only border nodes make use of them to establish intercluster keys as follows.

Key Establishment. Every border node u of cluster i is given a series of pairs $\langle v, j\rangle$ for border node v of cluster j after having received messages as (3). It can produce intercluster key $K^{uv} = f_{(i,j)}(u \oplus v)$ for each $\langle v, j\rangle$. Simultaneously, v also computes the same key, differently named $K^{vu} = f_{(j,i)}(v \oplus u)$ for u of i, for the symmetry as $(i, j) = (j, i)$. As soon as computing all the required intercluster keys, u erases all the given $(N_c - 1)$ keys.

2.5. Secure Transmission. Now, we present how these established keys practically secure in-cluster, intercluster, and individual communication as in (5) to (7), respectively:

$$u \longrightarrow \frac{v}{*} : t^u, \{p\}_{O^u}, [p]_{O^u}, \qquad (5)$$

$$u \longrightarrow v : \{p\}_{K^{uv}}, [p]_{K^{uv}}, \qquad (6)$$

$$\text{BS} \longrightarrow v : \{u \| i\}_{K^v}, [u \| i]_{K^v}, \{p\}_{K^u}, [p]_{K^u}. \qquad (7)$$

(i) Within a cluster, node u sends another node v or broadcasts packet p by transmitting the p's encryption and MAC by its current O^u with the current t^u as (5). Any receiver privately obtains p after deriving the O^u with the previously known b^u and the just arrived t^u.

(ii) Node u of cluster i always uses intercluster key K^{uv} to encrypt packet p and produce its MAC when it sends p to inter-neighbour v as in (6). As v also holds $K^{vu} = K^{uv}$, p is safely restored by v.

(iii) When BS individually informs node u in cluster i of packet p, it transmits p to the first node v on a given path as in (7). Whereas the former two to address the destination are repeatedly decrypted and reencrypted by the on-path nodes including v through a deal of in- and intercluster communication until they reach u, the rest two are just carried during the transmission and only decrypted by u. The opposite case of u to BS reverses the course we have described.

2.6. Handling Node Addition. Before its deployment, new node u is preloaded its individual key K^u only shared with BS, cluster ID i, unique base b^u, and all of i's member ids as assumed. BS has enough time to inform i's members of u's joining and random key K^{ui}, temporarily used for u's key

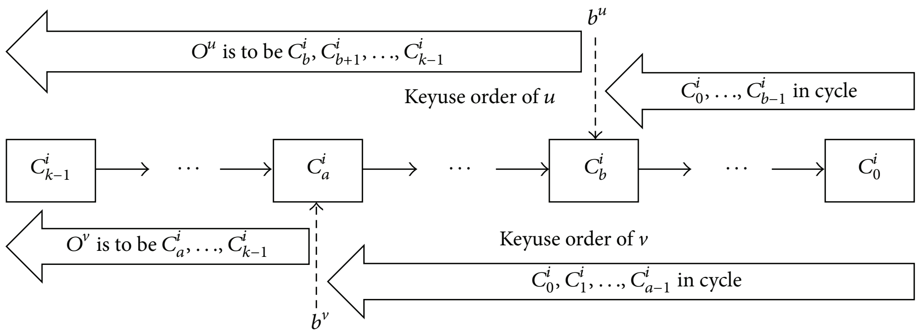

b^u
O^u is to be $C^i_b, C^i_{b+1}, \ldots, C^i_{k-1}$
Keyuse order of u
$C^i_0, \ldots, C^i_{b-1}$ in cycle
C^i_{k-1}
C^i_a
C^i_b
C^i_0
O^v is to be $C^i_a, \ldots, C^i_{k-1}$
Keyuse order of v
$C^i_0, C^i_1, \ldots, C^i_{a-1}$ in cycle
b^v

FIGURE 2: Different key-use orders of nodes u and v on their shared key chain $\{C^i_l\}$ for $0 \le l < k$. The periodically updated one-time encrypting key of u is given by $O^u = C^i_{(b^u + t^u \bmod k)}$, where b^u is the base of u and its packet sequence number t^u increases by 1 from 0. This is the same for v with its own b^v.

chain acquisition, in advance by letting i's head propagate the information. The notification is started as in (7), where $u||i$ and p are, respectively, replaced by $head||i$ and $u||K^{ui}$. Then, u first exchanges messages with its neighbours as in (8), and then (9) or (3) to obtain their bases and $\mathbf{C}^i$, $(k-1)$ recursions of $h(C^i_{k-1})$:

$$u \longrightarrow * : u, i, \{b^u\}_{K^{ui}}, [b^u]_{K^{ui}}, \tag{8}$$

$$v \longrightarrow u : v, \left\{C^i_{k-1}||b^v\right\}_{K^{ui}}, \left[C^i_{k-1}||b^v\right]_{K^{ui}}. \tag{9}$$

For every pair $\langle v, j\rangle$ for interneighbour v in cluster j if exists, u privately loads its generated intercluster key $K^{uv} = f_{K^{ui}}(u \oplus v)$ via a secure u-to-v path. As we assume that this entire node joining procedure is finished in T_{add}, the temporal key K^{ui} is automatically dropped by every member after T_{add} passes from when it is given.

2.7. Handling Node Eviction. A node is regarded to be evicted when its battery seems to be exhausted; when a large portion of its communication links do not work; or when it is detected as captured. Any of its neighbouring nodes perceiving one of the conditions announces it to the entire network. As soon as its uselessness is notified, its cluster members and interneighbours discard every related secret from their memory. In particular, if the node is captured as in the last case, the rest of its cluster should update the shared key chain as in the rechaining and key chain distribution. Also, its in-neighbours individually reselect and broadcasts a new base different from the previous one to their in-neighbours as in (1).

3. Analysis of Our Framework

In this section, we, in turn, analyse our framework in the following performance metrics: network connectivity, resiliency against impersonating attacks by node capture, and resource requirements.

3.1. Connectivity. Because of its deterministic nature, our framework achieves perfect connectivity between any two neighbouring nodes for both in-cluster and intercluster communication at the initialisation phase. This also holds for newly added nodes due to the prior node ID announcement by BS as in Section 2.6.

3.2. Resiliency. Since we view that active attacks, such as false data injection, most degrade network performances, we make the following strong attack model by node capture.

(i) The attacker can retrieve all the information stored in a sensor node once it captures the sensor node.

(ii) The attacker can capture a set of sensor nodes selectively in a WSN.

(iii) The attacker ultimately aims at impersonating legitimate nodes to inject false data with compromised keys.

An attacker can purposely locate and capture sensors having more secrets as border nodes in our framework by selectively attacking such nodes. Such an attacker can impersonate only *existing nodes* whose ids are compromised because a new node with a falsified ID is thoroughly excluded due to the prior node ID announcement by BS. Thus, the resiliency against this attack is measured by estimating the fraction of total sensor nodes that properly impersonate id-known nodes by an attack, modeled as $\{(c_i, \alpha_i, \beta_i)\}$ for every cluster i in a network of N nodes and N_c clusters, where c_i is the number of i's captured nodes, α_i is the distinction rate amongst c_i's in-neighbours in $(0, 1]$, and β_i is the bordering ratio of c_i in $[0, D_b/D_c]$. In this attack model, the greater α_i or β_i, the higher the selectiveness of the attack. Then, we formulate the overall impersonated fractions on in-cluster, intercluster, and individual communication, p^*_{in}, p^*_{int}, and p^*_{ind}, respectively, as follows:

$$\begin{gathered} p^*_{\text{in}}\left(\{(c_i, \alpha_i)\}\right) = \frac{1}{N_c}\sum_{i=1}^{N_c} p_{\text{in}}\left(c_i, \alpha_i\right), \\ p_{\text{in}}\left(c_i, \alpha_i\right) \\ \le \frac{c_i + \left(\alpha_i c_i D_n \left(D_n - 1\right) / \left(D_c - 1\right) k\right) + \left(\left(D_c - \alpha_i c_i D_n - 1\right) D_n / D_c k^2\right)}{D_c}, \end{gathered} \tag{10}$$

$$p^*_{\text{int}}\left(\{(c_i, \beta_i)\}\right) = \frac{1}{N_c}\sum_{i=1}^{N_c} \frac{c_i \beta_i}{D_b}, \tag{11}$$

$$p^*_{\text{ind}}\left(\{c_i\}\right) = \frac{\sum_{i=1}^{N_c} c_i}{N}. \tag{12}$$

The overall p^*_{in} is the normalised sum of impersonated fractions over every cluster, p_{in}, given by (10). In cluster i, given (c_i, α_i), adversary a behaves as follows:

(i) a completely impersonates every node u of c_i nodes,

(ii) to impersonate each of u's $\mathcal{O}(D_n)$ in-neighbours with their compromised bases, a should speculate its other $(D_n - 1)$ in-neighbours than u by $(D_n - 1)/(D_c - 1)$ as well as their sequence numbers by $1/k$, for every u,

(iii) to impersonate each of the rest $(D_c - \alpha_i c_i \mathcal{O}(D_n) - 1)$ members, a should speculate its $\mathcal{O}(D_n)$ neighbours by D_n/D_c, their bases by $1/k$, and sequence numbers by $1/k$,

The three terms of the numerator of (10), respectively, represent each case given above. For intercluster and individual communication, we can simply count and normalise as in (11) and (12) since one captured node reveals only its individual key and additionally intercluster keys if it is a border node. We simulate our resiliency levels for the different communication patterns with those of alternatives given different key chain lengths in Section 4.2.

3.3. Resource Requirements

Storage. Because the key size is usually larger than any other secret as a node ID or a base, we discuss only the number of retained keys for storage overhead. After the initialisation, every node obtains a single individual key, k, in-cluster keys of its key chain and additionally $\mathcal{O}(D_n)$ intercluster keys if it is a border node. Thus, the required storage of our framework is mainly due to the key chain length k given D_n.

Communication. The communication overhead for our security framework occurs between a neighbouring pair during initial keying, rekeying, and keying for a new node. At the initialisation, the pair exchanges only their ids and bases, whereas a seed key for a new key chain and new bases travel for re-keying. For a new node, the pair where one is the new node exchanges at most their ids, bases, a temporal or intercluster key, and the last key of the currently shared key chain as in Section 2.6.

Computation. In the initialisation, every node does $\mathcal{O}(D_n)$ times of MAC to securely obtain its in-neighbours' bases and additionally $\mathcal{O}(D_n)$ times of PRF f to generate intercluster keys if it is a border node. To update the key chain, it restores the sent seed key by one MAC operation and generates a new key chain of k recursions of OWF h. Given a new node, its neighbours can verify its base or sent intercluster key by one MAC operation. The new node operates one MAC to extract the last key and $(k - 1)$ recursions of h to generate its needed key chain. Letting r be the key size in bit, OWF $h : \{0,1\}^r \rightarrow \{0,1\}^r$ consumes $O(r)$ computation, whereas its inverting cost is $O(2^r)$ [17]. PRF $f : \{0,1\}^r \rightarrow \{0,1\}^r$ also has a similar overhead [18]. Since MAC is usually regarded as a kind of hash function as OWF [1], MAC has a similarly low computation complexity as well.

4. Comparison with Previous Studies

In this section, we compare the performances of our proposed framework with those of the following three selected conventional studies.

2KP (see [8]). BS has two key pools of M_1 and of M keys and assigns two key sets, k_1 keys from the M_1-pool and k keys from the M-pool, to each node in advance. As soon as every node is deployed, it broadcasts the key ID sequence of its k_1-key set. Only pairs sharing one or more common keys can establish a pairwise key somehow computed by a PRF, and then every node drops its k_1-key set. For node addition, a new node and its neighbours exchange the key ID sequences of their k-key sets to establish their pairwise keys as before.

LOCK (see [19]). This utilises two layers of keys for clustered WSNs. BS communicates only with weakly trusted cluster key servers (KSes) with $(k_b + m_b)$ keys by assigning a unique subset of k_b keys from ${}_{(k+m)}C_k$ combinations to each node before deployment. Every KS distributes its generated $(k+m)$ keys amongst its cluster members in the same manner. In both layers, each member establishes pairwise keys within its included group by exchanging their key ids. Additionally, every regular node shares k' backup keys with BS to report the compromise of its KS.

LEAP+ (see [1]). Initially, every node u is preloaded the same set of e keys, termed $K^1, K^2, \ldots, K^e$, by which u can derive its own base key in session l by $K^l_u = f_{K^l}(u)$. For u to establish pairwise key with every neighbouring v in session l, it first broadcasts its ID and waits for the encrypted v's ID with K^l_u presumed by v. Then, u derives K^l_v as well and the pair individually computes the pairwise key by $K^l_{uv} = f_{K^l_v}(u)$. After the session ends, every node u erases K^l and K^l_u from its memory. Every node is also preloaded a unique one-way key chain of k keys and transmits the last key as the current one-time key to its neighbours before it first attempts to broadcast. Then, every time it broadcasts a message, it uses the current one-time key to encrypt both the message and the next one-time key. The current one-time key is discarded after it is used in encryption or decryption.

In the following simulations, we vary only the number of used or stored keys, k_1, k, k_b, m, and m_b, to see its impact while fixing the network parameters as $N = 10000$, $N_c = D_c = 100$, $D_n = 50$, and $D_b = 30$.

4.1. Connectivity.

As already stated, the deterministic key establishing methods in ours and LEAP+ guarantee perfect connectivity regardless of the number of keys.

Since 2KP has stably high resiliency with 200 selected keys from 10000 keys [8], we take that $k_1 = 200$, $k = \{50, 200\}$, and $M_1 = M = 10000$. To see the cases with and without new nodes, we also consider $(N_1, N_2) = \{(10000, 0), (9000, 1000)\}$, where N_1 and N_2 are, respectively, the numbers of initially deployed nodes and of newly added nodes such that $N = N_1 + N_2$. Respectively saying cp_1 and cp_2, the connecting

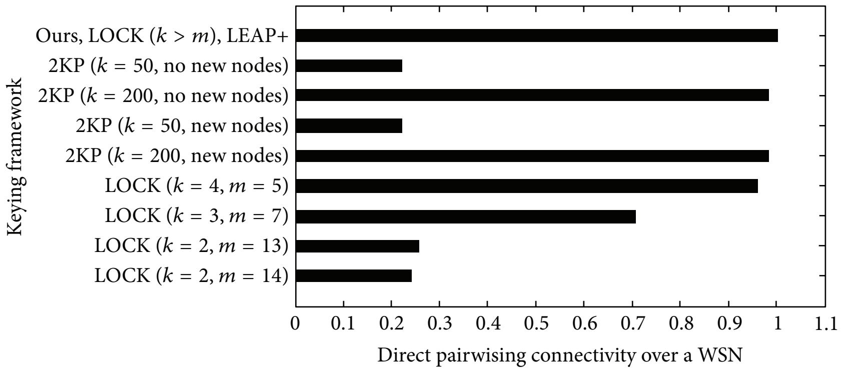


Figure 3: Comparison of direct pairwising connectivity amongst ours, 2KP, LOCK, and LEAP+.

probabilities between any neighbouring pair after N_1 nodes, are deployed and the one after N_2 nodes are added we formulate the overall connectivity of 2KP, cp_{2KP}, as follows:

$$cp_{2KP} = \frac{N_1 cp_1 + N_2 cp_2}{N_1 + N_2}, \qquad (13)$$

$$cp_1 = 1 - \frac{{}_{(M_1-k_1)}C_{k_1}}{{}_{M_1}C_{k_1}},$$

$$cp_2 = 1 - \frac{{}_{(M-k)}C_k}{{}_{M}C_k}. \qquad (14)$$

In both (13) and (14), the second term of the right represents the probability that two nodes do not share any keys to form a secure connection.

Regarding that $k = k_b$ and $m = m_b$, the connectivity of LOCK, cp_{LOCK}, is generally given in two different cases as follows:

$$cp_{\text{LOCK}} = \begin{cases} 1 & \text{if } k > m \\ 1 - \dfrac{{}_{m}C_k}{{}_{(k+m)}C_k} & \text{if } k \leq m. \end{cases} \qquad (15)$$

The upper equation holds by the fact that one's key set has at least one common key with every other's key set if $k > m$, whereas the key sharing probability for $k \leq m$ is given as cp_1 or cp_2 by its probabilistic nature. For $k \leq m$, we take that $(k, m) = \{(4, 5), (3, 7), (2, 13), (2, 14)\}$ to, respectively, offer 126, 120, 105, and 120 key combinations for storage efficiency in a 100-node cluster.

The comparison of direct pairwising connectivity amongst ours, 2KP, LOCK, and LEAP+ has given our setting is resulted as in Figure 3. As illustrated, our framework, LEAP+ and LOCK for $k > m$, has the perfect connectivity. The connectivity of 2KP is highly sensitive with k regardless of node addition. LOCK is more likely to share keys as the difference from m to k is smaller. Thus, we observe that our framework enhances network connectivity higher than 2KP and LOCK of $k \leq m$ regardless of the numbers of keys and of newly added nodes.

4.2. Resiliency. If c nodes are captured in total, no nodes other than the captured nodes can be impersonated in 2KP and LEAP+. Since they do not keep the keys based on which neighbouring pairs establish their pairwise keys using a PRF, the pairwise keys are hardly discovered by unauthorised parties due to the randomness of PRF. For broadcasting in LEAP+, every node owns its one-way key chain as well as its neighbours' one-time broadcasting keys. Even though the attacker obtains such a broadcasting key, it is hard to derive its future broadcasting keys by the one wayness of OWF [17]. Similarly, our impersonated probability on individual communication is given by c/N as well.

Table 1: Storage overhead comparison amongst ours, 2KP, LOCK, and LEAP+ in bit.

Framework	Node type	Keys
Ours	Nonborder	$\mathcal{O}(rk)$
	Border	$+\mathcal{O}(rD_n)$
2KP		$\mathcal{O}(r(k + D_n))$
LOCK	KS	$\mathcal{O}(r(k_b + k + D_n))$
	Regular	$\mathcal{O}(r(k + k' + D_n))$
LEAP+		$\mathcal{O}(r(e + k))$

In LOCK, k keys are always kept to establish pairwise keys with new nodes even though the keys are periodically updated. The fraction of total sensor nodes that is impersonated by $c = \sum_{i=1}^{N_c} c_i$ captured nodes, p_{LOCK}, is given as follows:

$$p_{\text{LOCK}}(\{c_i\}) \geq \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{A_i}{{}_{(k+m)}C_k}, \qquad (16)$$

$$A_i = \min {}_{\check{k}_i}C_k \geq c_i \quad \text{for } k \leq \check{k}_i \leq k + m.$$

We say that A_i is the minimum number of key combinations in cluster i that c_i captured nodes can restore. In other words, the attacker obtains at least $\check{k}_i$ distinct keys, which produces ${}_{\check{k}_i}C_k$ key combinations, from kc_i compromised keys. This means that at least ${}_{\check{k}_i}C_k$ nodes can be impersonated until they are detected as compromised.

Given c total captured nodes in $[500, 3000]$ for c_i in $[0, D_c]$ for every cluster i, the average resiliency comparison work over 100 simulations is shown as in Figure 4. More specifically, we consider high selective attacks for α_i in $[0.7, 1]$ and β_i

Table 2: Communication overhead comparison amongst ours, 2KP, LOCK, and LEAP+ in bit.

Framework	Initial keying	Rekeying	Keying for a new node
Ours	$\mathcal{O}(r)$	$\mathcal{O}(r)$	$\mathcal{O}(r)$
2KP	$\mathcal{O}(rk_1)$	N/A	$\mathcal{O}(rk)$
LOCK	$\mathcal{O}(rk)$	$\mathcal{O}(rk)$	$\mathcal{O}(rk)$
(KS)	$+\mathcal{O}(rk_b)$	$+\mathcal{O}(rk_b)$	N/A
LEAP+	(Pairwise keying) $\mathcal{O}(r)$	(Whenever broadcasting) $\mathcal{O}(r)$	(Pairwise keying) $\mathcal{O}(r)$

in $[0.15, 0.3]$ for our framework and the perfectly connected cases of LOCK as $(k, m) = \{(5, 4), (7, 3), (13, 2), (14, 2)\}$. Whereas the impersonated probabilities of ours, 2KP, and LEAP+ are under or around c/N for every c, LOCK has weaker resiliency on average with any (k, m). Thus, we observe that, regardless of the number of retained keys, our proposed framework similarly works as 2KP and LEAP+ known for the strongest resiliency do against impersonating by node capture.

4.3. Resource Requirements. Now, we compare our framework with 2KP, LOCK, and LEAP+ in three resource overhead metrics: storage, communication, and computation. Every analysed overhead is represented by the big O notation to see its maximum complexity even in the worst case.

Storage. Table 1 provides the storage overheads of the selected keying frameworks. Regarding that r stands for the size of key in bit, the different storage requirements to nonborder and to border nodes in our framework are given as stated in Section 3.3. By 2KP, every node is preloaded k keys from the M-pool for new nodes and shares pairwise keys with $\mathcal{O}(D_n)$ neighbours. In LOCK, every KS keeps k_b keys for the BS-KSes communication and k keys for its cluster communication, whereas every regular node stores k' keys to communicate with BS as well as k keys as its KS does. Any group member establishes pairwise keys with its $\mathcal{O}(D_n)$ neighbours whatever the group is. LEAP+ initially assigns every node e keys to establish pairwise keys with any nodes added in any session l $(\leq e)$ and a k-length key chain to broadcast. Both types of keys are dropped just after they are used. Usually, k of 2KP and e and k of LEAP+ take larger values for high connectivity and network longevity, respectively. Thus, on the storage overhead, our framework is superior to 2KP and LEAP+ but is not to LOCK in our simulations, where the cases of LOCK with smaller k and k_b achieve better resiliency than those with large k and k_b as in Figure 4.

Communication. While assuming that the key size, r, is greater than node ids, key ids, bases, and the MACs of all of these without losing generality, we present the communication overheads required between a neighbouring pair for each of initial keying, rekeying,and keying for a new node by the different studies as in Table 2. For any keying course, our keying framework consumes communication sources with a single key with its MAC, two bases with their MACs, or two node ids as discussed in Section 3.3. Similarly, a neighbouring pair exchanges associated node ids or a broadcasting key with its MAC in LEAP+. By contrast, in 2KP and LOCK, every node broadcasts the sequence of their own key ids as needed to find neighbours having common keys. Since the broadcasting key delivery of LEAP+ more often occurs than our rechaining, our framework reduces the communication overheads of all the keying phases over 2KP, LOCK, and LEAP+.

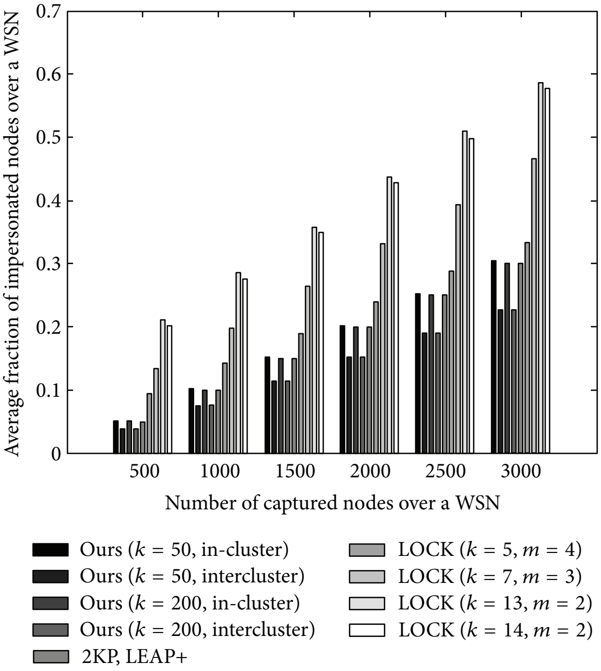


Figure 4: Comparison of average resiliency against impersonating by node capture amongst ours, 2KP, LOCK, and LEAP+.

Computation. All the complexities required to chain keys by an OWF, to produce an MAC by a hash function, and to generate a pairwise key by a PRF can be regarded to be negligible [1, 8]. Thus, the security frameworks without the key ID comparison, ours and LEAP+, have lower computation overheads than 2KP and LOCK do.

Although our framework does not achieve the least storage overhead, it is fairly competitive because the resource consumption of wireless sensor nodes is usually dominated by communication [20].

5. Conclusion

In the paper, we have proposed a new dynamic keying framework for large-scale clustered WSNs, widely employed to implement ubiquitous sensor networks. In the framework, different keying mechanisms, respectively, not only protect in-cluster, intercluster, and individual communication but also effectively handle node addition and eviction. Our proposed key-use ordering mechanism of a cluster-shared one-way key chain, illustrated in Figure 2, and intercluster key establishment using the preloaded key matrix achieve perfect connectivity as well as well protect wireless sensors from impersonating by node capture with low resource overheads. Such our claims have been discussed by the given theoretic analyses and varied simulations. As one of extensions for this work, we may raise energy efficiency and practicability by utilizing environment energy and considering crosslayer networking as in [21].

Acknowledgments

This work was supported by the Basic Science Research Program through a Grant from the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (2012-0005793 and 2011-0013776). This work was also supported by the IT R&D program of MKE/KEIT (10039165, development of learner-participatory and interactive 3D virtual learning contents technology).

References

[1] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: efficient security mechanisms for large-scale distributed sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 4, pp. 500–528, 2006.

[2] T. Kavitha and D. Sridharan, "Security vulnerabilities in wireless sensor networks: a survey," *Journal of Information Assurance and Security*, vol. 5, pp. 31–44, 2010.

[3] H. Yang, F. Ricciato, S. Lu, and L. Zhang, "Securing a wireless world," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 442–453, 2006.

[4] J. Zhang and V. Varadharajan, "Wireless sensor network key management survey and taxonomy," *Journal of Network and Computer Applications*, vol. 33, no. 2, pp. 63–75, 2010.

[5] C. Y. Chen and H. C. Chao, "A survey of key distribution in wireless sensor networks," *Security and Communication Networks*, 2011.

[6] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, New York, NY, USA, 1997.

[7] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TELSA broadcast authentication protocol," Tech. Rep., Carnegie Mellon University, Pittsburgh, Pa, USA, 2005.

[8] A. K. Das, "An efficient random key distribution scheme for large-scale distributed sensor networks," *Security and Communication Networks*, vol. 4, no. 2, pp. 162–180, 2011.

[9] M. Corporation, Tmote Sky: ultra low power IEEE 802. 15. 4 compliant wireless sensor module humidity, light, and temperature sensors with USB, 2006.

[10] D. Djenouri, L. Khelladi, and A. N. Badache, "A survey of security issues in mobile ad hoc and sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 7, no. 4, pp. 2–28, 2005.

[11] G. Jeong and H. S. Yang, "Efficiently secure image transmission against tampering in wireless visual sensor networks," in *Proceedings of the 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS '11)*, pp. 172–177, September 2011.

[12] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 586–597, March 2004.

[13] L. Ying, R. Srikant, D. Towsley, and S. Liu, "Cluster-based back-pressure routing algorithm," *IEEE/ACM Transactions on Networking*, vol. 19, no. 6, pp. 1773–1786, 2011.

[14] N. A. Alrajeh, S. Khan, and B. Shams, "Intrusion detection systems in wireless sensor networks, a review," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 167575, 7 pages, 2013.

[15] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[16] H. Tan, J. Zic, S. Jha, and D. Ostry, "Secure multihop network programming with multiple one-way key chains," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 16–31, 2011.

[17] O. Goldreich, "Candidate one-way functions based on expander graphs," in *Studies in Complexity and Cryptography*, vol. 6650 of *Lecture Notes in Computer Science*, pp. 76–87, Springer, Berlin, Germany, 2011.

[18] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*, Cambridege University Press, Cambridege, UK, 2009.

[19] M. Eltoweissy, M. Moharrum, and R. Mukkamala, "Dynamic key management in sensor networks," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 122–130, 2006.

[20] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor," in *Proceedings of the ACM 1rst International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 1–13, November 2003.

[21] N. A. Alrajeh, S. Khan, J. Lloret, and J. Loo, "Secure routing protocol using crosslayer design and energy harvesting in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 374796, 11 pages, 2013.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 391986, 8 pages
http://dx.doi.org/10.1155/2013/391986

Research Article

Key Schemes for Security Enhanced TEEN Routing Protocol in Wireless Sensor Networks

Saewoom Lee,[1] Youngtae Noh,[2] and Kiseon Kim[1]

[1] *School of Information and Mechatronics, Department of Nanobio Materials and Electronics (DNE) World-Class University (WCU), Gwangju Institute of Science and Technology (GIST), 1 Oryong-dong, Buk-Gu, Gwangju 500-712, Republic of Korea*
[2] *Department of Computer Science, University of California, Los Angeles (UCLA), Los Angeles, CA 90095, USA*

Correspondence should be addressed to Youngtae Noh; ytnoh@cs.ucla.edu

Received 7 March 2013; Accepted 25 May 2013

Academic Editor: S. Khan

In wireless sensor networks (WSNs), hierarchical routing protocol is commonly used for energy efficiency. In particular, the TEEN (Threshold sensitive Energy Efficient sensor Network) protocol is used widely as a basic clustered multihop routing protocol. However, energy efficient routing protocols without proper security suffer from many security vulnerabilities. Hence, in this paper, we propose a hybrid key scheme specially for the TEEN protocol: a symmetric key scheme for the intracluster and a public key scheme for the intercluster. The simulation results show that network lifetime of the proposed hybrid key scheme decreases about 8% than the TEEN protocol and about 4% compared with the TEEN protocol with symmetric key scheme. On the other hand, a hybrid key scheme provides better probability of successful transmission than that of the symmetric key scheme.

1. Introduction

Wireless sensor networks (WSNs) can be applied to various applications such as safety monitoring of special spaces and buildings, traffic monitoring, environmental pollutant tracking, ocean and wildlife monitoring, home appliance management, and many military applications. In WSNs, one of the most significant constraints is the limited battery power of the nodes. Since the randomly deployed nodes are infeasible to be recharged. In this regard, it is worthwhile to pursue energy efficiency in WSNs. To acquire energy efficiency, various routing protocols are proposed in the literature such as SPIN, LEACH, and TEEN [1, 2]. Among the classified routing protocols as flat, hierarchical, and location-based routing protocol, the hierarchical routing protocol is proper in the view of energy efficiency due to the cluster head (CH) which performs data aggregation from non-CH nodes and directly communicates with the base station (BS). In particular, the TEEN (Threshold sensitive Energy Efficient sensor Network) protocol [3] is a basic routing protocol of hierarchical clustered multihop routing protocol. However, most routing protocols including the TEEN protocol in WSNs assume a trusted environment where all sensor nodes cooperate each other without any attacks. As a result, routing protocols suffer from many security vulnerabilities like as Denial of Service (DoS), injecting, and impersonating. Thus, an attacker can make the network useless [4, 5]. Therefore, designing a secure routing protocol is necessary for WSNs to provide secure data transmission regardless of opponent activities.

To cope with these problems, most security protocols are based on cryptographic operations that involve keys. Two types of key schemes are used in cryptography generally. The first one is the symmetric key scheme, which is computationally inexpensive, and can be used to achieve some of security goals. However, one major drawback with this scheme is the key exchange problem; that is, the two communication nodes must somehow know the shared key before communicating securely. Unfortunately, if an attacker can capture the symmetric key, the whole network can be broken because the attacker can decrypt every encrypted data by using the symmetric key [6, 7]. The other type of key scheme is public key scheme, which uses a pair of keys (p, q) where p means the public key and q indicates the private key corresponding to the public key p. Different from the symmetric key scheme, only the receiver can decrypt the encrypted data in the public key scheme [8]. So public key scheme allows for flexible key management but requires a significant amount of computation due to the complex algorithm.

In this paper, we propose a hybrid key scheme that uses both symmetric and public key schemes in order to take the advantage of the rapid calculation times of the symmetric key scheme and flexible key management of the public key scheme. By considering a hybrid key scheme, we can offer security into the TEEN protocol. Usually, the encrypted data is transmitted from the sending node, and then it decrypts at the receiving node. In hybrid key scheme, two different key schemes are used depending on the types of communication for the TEEN protocol. That is, the symmetric key scheme is applied to intracluster communication, and the public key scheme is used for intercluster communication. Also, to provide the assurance of the identities among communication nodes, we create hashed value generated form the hash function. This hashed value is used to authenticate the origin of the messages as a message authentication code (MAC).

The rest of the paper is organized as follows. In Section 2, we briefly overview the TEEN protocol to explain the background for reactive routing protocol and related works. In Section 3, we describe the hybrid key scheme adapted to the TEEN protocol for the security of WSNs. In Section 4, we simulate a hybrid key scheme to evaluate the performance. We describe analysis of the security of the protocol in Section 5. Finally we conclude the paper in Section 6.

2. TEEN Protocol and Related Works

In this section, we briefly introduce the TEEN protocol and several routing protocols connected with the TEEN protocol for WSNs.

There are routing protocol groups based on their mode of functioning and the type of target application in WSNs: proactive and reactive routing protocols. In proactive routing protocol, once the cluster heads (CHs) are decided after cluster exchanging, the CH node creates a TDMA schedule and assigns each node a time slot when it can transmit. After setup phase, cluster members sense the phenomena and transmit the data to the CH. The CH aggregates this data and sends aggregated data to the higher level CH, or the BS depends on the network hierarchy. Low-Energy Adaptive Clustering Hierarchy (LEACH) [9] is a good example of a proactive routing protocol with some small differences.

On the other hand, the CH broadcasts in the following threshold values to its cluster members at every cluster setup phase in the TEEN protocol [3] which is the most typical protocol for reactive routing protocol.

Hard threshold (H_T): it is an absolute value for the sensed attribute. If the node senses this value, it turns on its transmitter and reports the data to the CH.

Soft threshold (S_T): it is a small variation in the value of the sensed attribute which causes the node to turn on its transmitter.

The nodes sense their surroundings continuously. The first time a sensed data reaches its hard threshold value, the node transmits the sensed data. The sensed value is stored to a variable called *sensed value* (SV). The node will transmit the data in current round only when both the following conditions are true:

(1) the sensed data is greater than the hard threshold,

(2) the sensed data differs from SV by an amount equal or greater than the soft threshold.

Thus, the hard threshold tries to reduce the number of transmission by sending only when the sensed data is in the range of interest. Also, the soft threshold reduces the number of transmission by excluding from the transmissions which have little or no change in the sensed data.

The TEEN protocol is succeeded by the APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient Sensor Network) [10] protocol which aims at capturing periodic data aggregations and respond to time-critical events. In APTEEN protocol, once the CHs are decided, the CH broadcasts four parameters: attributes, thresholds, TDMA schedule and count time. By using these parameters, the APTEEN protocol can provide access to periodic data as well as be informed of events of certain significance. Also, the THCHP (Two-level Hierarchical Clustering based Hybrid routing Protocol) [11] is expanded from the APTEEN protocol. The THCHP can be used for applications that require periodic data monitoring as well as warnings about critical events. The use of a two-level clustering hierarchy enables fixing the number of level 1 clusters K and optimizing the number of level 2 clusters so that the average sensor node energy dissipation is minimized.

Another modified version of the TEEN protocol is H-TEEN (Hierarchical Threshold sensitive Energy Efficient Network) [12] protocol where sensors self-organize into clusters and build a tree of transmissions, propagating data only to their parent in this tree. In H-TEEN protocol, it uses a constant number of hops to propagate the messages, having its cluster heads transmitting directly to the next level of the hierarchy. The H-TEEN protocol achieves high success rates in small area networks.

Kavitha and Viswanatha [13] have proposed Hybrid Reliable Routing (HRR) technique in wireless sensor networks. The HRR is intended to offer a hierarchical transmission environment by organizing randomly deployed sensor nodes into clusters efficiently. The remaining nodes can acquire the energy availability factor of the neighboring CHs. After that, they join that cluster which has more energy than other CHs, for that reason, ensuring service for a longer time. Once the CHs are identified, they generate a Dominating Set (DS). The members nodes of DS find least energy consumed multihop route to the sink. Meanwhile, graph theory can be used to generate the sensor clusters and help in identifying the CH.

As a basic routing protocol of hierarchical clustered multihop routing protocol, the TEEN protocol is succeeded by various routing protocols as we mentioned. However, all mentioned routing protocols are commonly focused on the fast data transmission with less energy consumption. Thus, they can be attacked from various security vulnerabilities to make the network useless. Therefore, the security for routing protocols including the TEEN protocol is necessary to provide secure data transmission.

3. Hybrid Key Scheme for the TEEN Protocol

In this section, we propose a hybrid key scheme adapted to the TEEN protocol.

Figure 1 introduces an example of hierarchical clustering routing protocols. Each cluster has a CH which aggregates data from cluster members. CH sends aggregated data to the BS or an upper level CH. These CHs, in turn, form a cluster with higher level CH as their CH. So some CHs can role as a second level CH. This action is repeated to form a hierarchy of clusters with the uppermost level cluster nodes for reporting directly to the BS. CHs at higher level in the hierarchical clustering need to send data over correspondingly larger distances.

As can be seen in Figure 1, there are three types of communication in the TEEN protocol: node-to-CH, CH-to-BS, and CH-to-CH communication. In node-to-CH communication case, a member node of the cluster tries to send the sensed data to the CH when it is satisfied the threshold values. On this occasion, a similar data can be transmitted by its neighbor nodes with high probability. In other words, if one similar data was damaged by an attacker, it can be restored by data of neighbor nodes. On the other hand, a CH, which has aggregated data of the cluster, sends whole cluster data to the BS or upper level CH. At this time, if a transmitted data was attacked by an attacker, a CH loses all information which sensed from its cluster or lower level cluster.

According to previous different types of communication, it needs different security method that depends on data integration to transmit the data securely. In node-to-CH communication case, even though sensed data which satisfies the thresholds can transmit, interested data can be detected by various nodes located around the event. Hence, a symmetric key scheme, which has simple algorithm and less calculation time, is adaptable for node-to-CH communication. On the other hand, in CH-to-BS and CH-to-CH communication cases, an aggregated data contains whole information of the cluster. If data modification or loss happens during communication, it is difficult to restore an information of the cluster. To prevent data loss or modification, a complex security algorithm is necessary, though a symmetric key scheme can give less calculation time. Therefore, a public key scheme, which has complex algorithm, is suitable for CH-to-BS and CH-to-CH communication. To sum up, it is appropriate to use different security method for different types of communication. That is, a hybrid key scheme is used for the TEEN routing protocol.

To apply the hybrid key scheme to the TEEN protocol, we append three procedures. One is addition of a symmetric key scheme inside the cluster, and the rest are addition of a public key scheme outside the cluster. Additionally, the last communication step is modified to use the hybrid key scheme during communication. To make secure keys for the hybrid key scheme, step of generating random numbers and exchanging those numbers are considered compared to the TEEN protocol. The next steps show the procedures of the hybrid key scheme for the TEEN protocol. Table 1 displays the notations used in this protocol description.

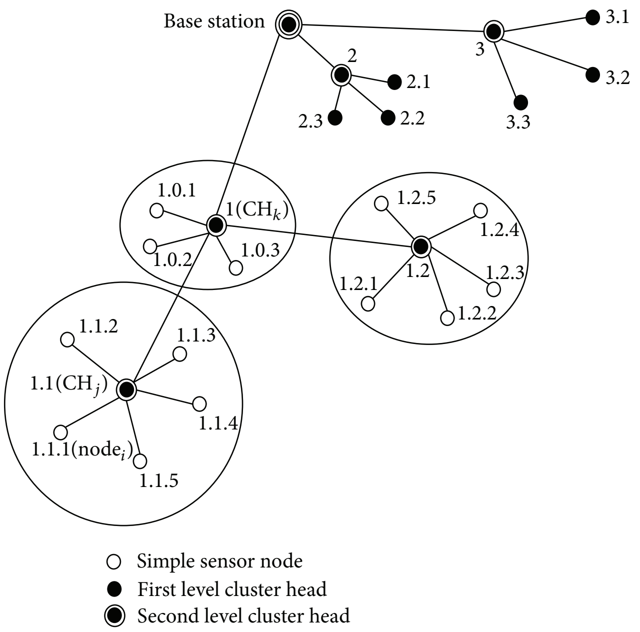


Figure 1: Hierarchical clustering routing protocol.

Table 1: Notations used in the protocol description.

Notation	Description
S	Generated random number before deployment
r_i	Random number generated at node i
$H(S)$	S is the data supplied to a hash function H
ID_j	Identification of node j
$A \parallel B$	Data A concatenated with data B
$E_K(A)$	Encrypt data A by using the key K
$D_K(C)$	Decrypt cipher C by using the key K
p_k	Public key used at node k (CH_k)
q_k	Private key corresponds to p_k at node k (CH_k)

Procedure 1. Figure 2 shows a process of symmetric key creation between node$_i$ and CH$_j$. First of all, insert a random number S into every node before node deployment. When a process of node deployment is finished, every node generates their own random number such as r_i for node$_i$. After generating a random number, node$_i$ and CH$_j$ create or exchange the following messages to make a shared key:

(1) node$_i$ and CH$_j$: $H(S) = k$,

(2) node$_i$ → CH$_j$: $E_k(\mathrm{ID}_i, r_i) \parallel \mathrm{ID}_i$,
$H[E_k(\mathrm{ID}_i, r_i) \parallel \mathrm{ID}_i]$,

(3) node$_i$ ← CH$_j$: $E_k(\mathrm{ID}_j, r_j) \parallel \mathrm{ID}_j$,
$H[E_k(\mathrm{ID}_j, r_j) \parallel \mathrm{ID}_j]$,

(4) node$_i$ and CH$_j$: $S^{r_i \times r_j}$.

In Message 1, each node calculates a hashed value k using a hash function and an inserted random number S. After CH selection and cluster setup steps, node$_i$ sends the join signal to be a member of CH$_j$. At this time, node$_i$ also sends

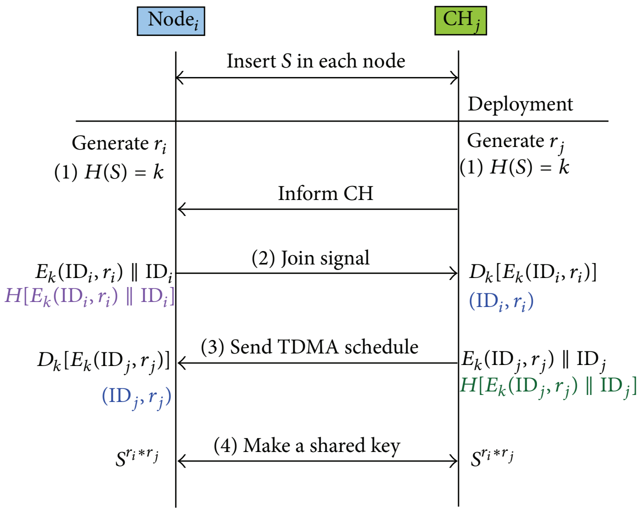

Node$_i$
CH$_j$
Insert S in each node
Deployment
Generate r_i
(1) $H(S) = k$
Generate r_j
(1) $H(S) = k$
Inform CH
(2) Join signal
$E_k(\text{ID}_i, r_i) \parallel \text{ID}_i$
$H[E_k(\text{ID}_i, r_i) \parallel \text{ID}_i]$
$D_k[E_k(\text{ID}_i, r_i)]$
(ID_i, r_i)
(3) Send TDMA schedule
$D_k[E_k(\text{ID}_j, r_j)]$
(ID_j, r_j)
$E_k(\text{ID}_j, r_j) \parallel \text{ID}_j$
$H[E_k(\text{ID}_j, r_j) \parallel \text{ID}_j]$
(4) Make a shared key
$S^{r_i * r_j}$
$S^{r_i * r_j}$

Figure 2: Process of symmetric key creation.

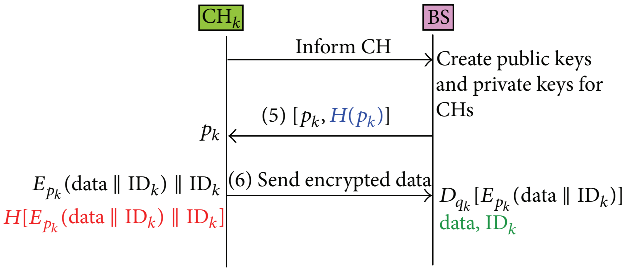

CH$_k$
BS
Inform CH
Create public keys and private keys for CHs
(5) $[p_k, H(p_k)]$
p_k
$E_{p_k}(\text{data} \parallel \text{ID}_k) \parallel \text{ID}_k$
$H[E_{p_k}(\text{data} \parallel \text{ID}_k) \parallel \text{ID}_k]$
(6) Send encrypted data
$D_{q_k}[E_{p_k}(\text{data} \parallel \text{ID}_k)]$
data, ID$_k$

Figure 3: Process of public key creation.

$E_k(\text{ID}_i, r_i) \parallel \text{ID}_i$ and $H[E_k(\text{ID}_i, r_i) \parallel \text{ID}_i]$, hashed value of $E_k(\text{ID}_i, r_i) \parallel \text{ID}_i$, for message authentication such as Message 2. Concatenated identification of node$_i$(ID$_i$) is used for confirming the sender. Then a CH$_j$ calculates a hashed value using delivered $E_k(\text{ID}_i, r_i) \parallel \text{ID}_i$. If a calculated hash value is same as the received hash value, we can use the message received from node$_i$. Else we just discard the whole messages. Through a message authentication and decryption, CH$_j$ can get an ID$_i$ and a random number r_i. In Message 3, the CH$_j$ sends a TDMA schedule to node$_i$ as a member of CH$_j$ for removing the collision. Also, CH$_j$ sends $E_k(\text{ID}_j, r_j) \parallel \text{ID}_j$ and $H[E_k(\text{ID}_j, r_j) \parallel \text{ID}_j]$. If a received message is valid to node$_i$, it can get a r_j and an ID$_j$ after decryption. By using Messages 1, 2, and 3, they can exchange their own generated random numbers r_i and r_j, so, node$_i$ and CH$_j$ can create a symmetric key, $S^{r_i \times r_j}$, only for node$_i$-to-CH$_j$ communication. In this shared key, exchanged r_i and r_j are used for exponent part, and a S is used for base part to create the symmetric key.

Procedure 2. As can be seen, in the TEEN protocol, the role of the BS is just receiving data from the CHs. But, one additional duty is added to BS in the hybrid key scheme.

Figure 3 shows a process of public key creation between CH$_k$ and BS. They can make public key for CH$_j$ using the following messages:

(5) CH$_k$ ← BS: p_k, $H(p_k)$,

(6) CH$_k$ → BS: $E_{p_k}(\text{data}, \text{ID}_k) \parallel \text{ID}_k$,
$H[E_{p_k}(\text{data}, \text{ID}_k) \parallel \text{ID}_k]$.

When the CHs inform their selection to the BS directly, the BS makes public and private key pairs. After creation the pairs, generated public key p_k for the CH$_k$ and its hashed value $H(p_k)$ are delivered to CH$_k$ as Message 5. CH$_k$ calculates a hashed value of p_k. If a calculated value and a received value are identical, p_k can be used as a public key for CH$_k$. When CH$_k$ sends aggregated data of a cluster to the BS, it uses a p_k for the data encryption. After encryption, CH$_k$ transmits encrypted data, $E_{p_k}(\text{data}, \text{ID}_k) \parallel \text{ID}_k$, and a hashed value, $H[E_{p_k}(\text{data}, \text{ID}_k) \parallel \text{ID}_k]$ as Message 6. The BS calculates hash value of encrypted data to compare received hash value with calculated hash value. If two values are identical, the BS decrypts an encrypted data by using a private key q_k corresponding to a public key p_k: $D_{q_k}(E_{p_k}(\text{data}))$.

Procedure 3. Figure 4 shows a process of public key creation among CH$_j$, CH$_j$, and the BS. They can make public keys for CHs using the following messages:

(7) CH$_k$ ← BS: p_k, $H(p_k)$,

(8) CH$_k$ → BS: $E_{p_k}(\text{data}, \text{ID}_k) \parallel \text{ID}_k$,
$H[E_{p_k}(\text{data}, \text{ID}_k) \parallel \text{ID}_k]$,

(9) CH$_k$ ← BS: p_k, $H(p_k)$,

(10) CH$_k$ → BS: $E_{p_k}(\text{data}, \text{ID}_k) \parallel \text{ID}_k$,
$H[E_{p_k}(\text{data}, \text{ID}_k) \parallel \text{ID}_k]$.

When the CH$_j$ informs its selection to the BS through CH$_k$, CH$_k$ sends concatenated ID list, (CH$_j$ ‖ CH$_k$), to the BS. During the creation of public and private key pairs from the BS, it creates key pairs (p_j, q_j) for CH$_j$ and (p_k, q_k) for CH$_k$, respectively. After creation, LIST1 contains $(\text{CH}_k, p_k) \parallel (\text{CH}_j, p_j)$. This LIST1 and a hashed value of LIST1, $H(\text{LIST1})$, delivered to CH$_k$ as Message 7. CH$_k$ calculates a hashed value of LIST1. If a calculated value and a received value are same, CH$_k$ extracts its public key p_k from the LIST1 for communication between CH$_k$ and the BS. On the other hand, CH$_k$ also sends LIST2, (CH_j, p_j), and a hashed value of LIST2, $H(\text{LIST2})$, to lower level CH$_j$ as Message 8. CH$_j$ tries to calculate a hashed value of LIST2. If the calculated value is identical as the received value from CH$_k$, CH$_j$ can use a public key p_j for communication between CH$_j$ and CH$_k$. When CH$_j$ transmits aggregated data to the BS through CH$_k$, an encrypted data, $E_{p_j}(\text{data}, \text{ID}_j) \parallel \text{ID}_j$, and a hashed value, $H[E_{p_j}(\text{data}, \text{ID}_j) \parallel \text{ID}_j]$, are transmitted to CH$_k$ as Message 9. After calculation and comparison between received and calculated values, CH$_k$ appends its ID, ID$_k$, end of encrypted data as $E_{p_j}(\text{data}, \text{ID}_j) \parallel \text{ID}_j \parallel \text{ID}_k$ and makes hashed value of modified data like as $H[E_{p_j}(\text{data}, \text{ID}_j) \parallel \text{ID}_j \parallel \text{ID}_k]$. When the BS receives these data from CH$_k$ as Message 10, the BS calculates hash value of encrypted data to compare a received hash value with a calculated hash value. If two values are identical, the BS decrypts an encrypted data by using a private key

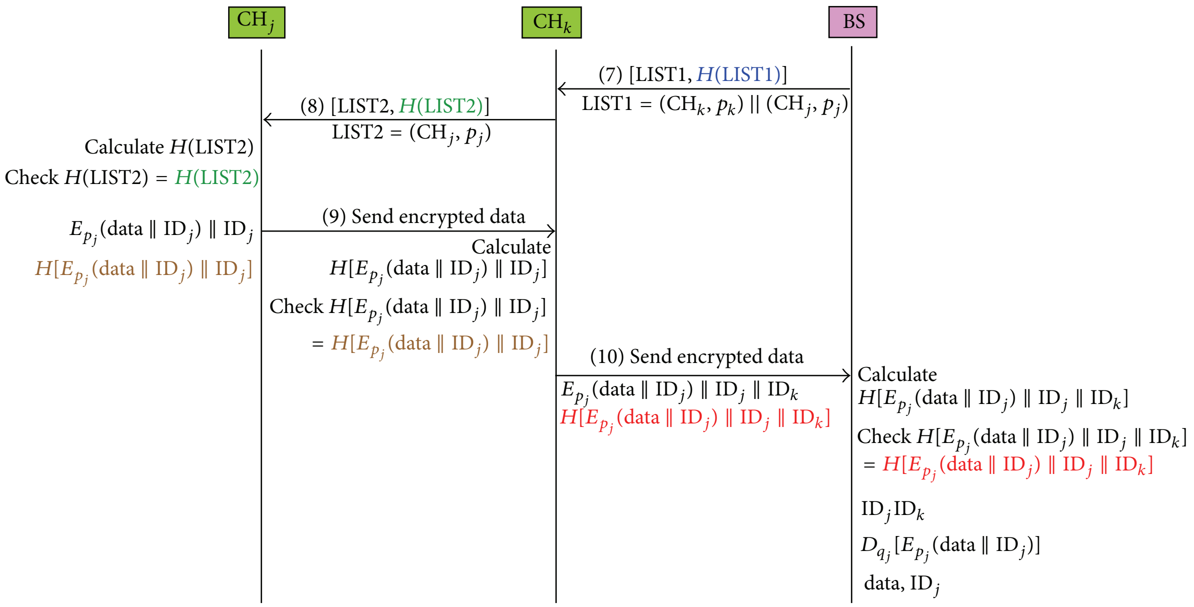

CH_j
CH_k
BS
(7) [LIST1, H(LIST1)]
LIST1 = $(\text{CH}_k, p_k) \| (\text{CH}_j, p_j)$
(8) [LIST2, H(LIST2)]
LIST2 = (CH_j, p_j)
Calculate H(LIST2)
Check H(LIST2) = H(LIST2)
(9) Send encrypted data
$E_{p_j}(\text{data} \| \text{ID}_j) \| \text{ID}_j$
$H[E_{p_j}(\text{data} \| \text{ID}_j) \| \text{ID}_j]$
Calculate
$H[E_{p_j}(\text{data} \| \text{ID}_j) \| \text{ID}_j]$
Check $H[E_{p_j}(\text{data} \| \text{ID}_j) \| \text{ID}_j]$
$= H[E_{p_j}(\text{data} \| \text{ID}_j) \| \text{ID}_j]$
(10) Send encrypted data
$E_{p_j}(\text{data} \| \text{ID}_j) \| \text{ID}_j \| \text{ID}_k$
$H[E_{p_j}(\text{data} \| \text{ID}_j) \| \text{ID}_j \| \text{ID}_k]$
Calculate
$H[E_{p_j}(\text{data} \| \text{ID}_j) \| \text{ID}_j \| \text{ID}_k]$
Check $H[E_{p_j}(\text{data} \| \text{ID}_j) \| \text{ID}_j \| \text{ID}_k]$
$= H[E_{p_j}(\text{data} \| \text{ID}_j) \| \text{ID}_j \| \text{ID}_k]$
$\text{ID}_j \text{ID}_k$
$D_{q_j}[E_{p_j}(\text{data} \| \text{ID}_j)]$
data, ID_j

FIGURE 4: Process of public key creation.

q_j corresponding to a public key p_j, because ID_j is attached previously than ID_k. In this procedure, when CH_k tries to send its own aggregated data to the BS, it uses simply same step as Procedure 2.

After Procedures 1, 2, and 3, a hybrid key scheme, which uses a symmetric key scheme for intracluster communication and a public key scheme for intercluster communication, is added to the TEEN protocol for secure data transmission.

4. Performance Evaluation

In this section, we explain the simulation results performed on NS-2 [14] about an energy consumption and a probability of transmission. The purpose of the simulation is to evaluate the improvement about a probability of transmission over that of the symmetric key scheme, while energy consumption closes to symmetric key scheme's energy consumption.

For the comparison, we simulated the TEEN protocol on NS-2. The scenario of our simulation is as follows: 100 nodes distributed randomly in a 100 m × 100 m area. Each sensor has an initial energy of 2 J, and a sensing area is 10 m. The time duration of each round is 20 seconds. On the other hand, AES-128 and XTR-128 are used for symmetric and public key schemes, respectively, because energy consumption of AES is smallest among different symmetric key schemes: 3DES, RC5, Blowfish, and so forth. Besides, energy cost of XTR is also smallest among different public key schemes: DH, ECDH, ECDSA, and so forth [8]. Moreover, we use the SHA-1 for message authentication. The whole simulation parameters are listed in Table 2.

4.1. Network Lifetime. Figure 5 shows the comparison of network lifetimes for the TEEN protocol and the TEEN protocol with symmetric, public, and hybrid key schemes.

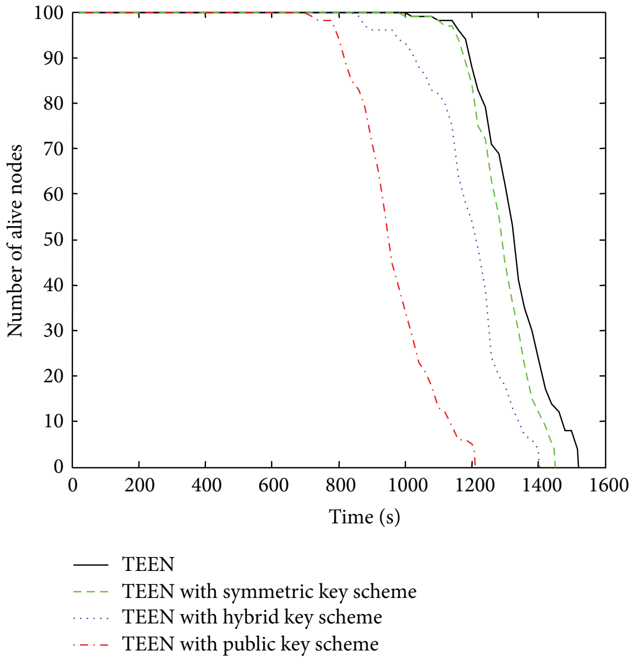


FIGURE 5: Number of alive nodes as a function of time, for the TEEN protocol without security and with symmetric, public, and hybrid key schemes.

As can be seen, when we use only a symmetric key scheme, network lifetime decreases about 5% compared with that of the TEEN protocol; on the other hand, network lifetime decreases about 21% when we use only the public key scheme. However, using the hybrid key scheme, network lifetime decreases about 8% compared with the TEEN protocol and about 4% compared with the TEEN protocol with symmetric

Table 2: Simulation parameters.

Monitoring area	100 m × 100 m
The number of nodes	100
Sensing range	10 m
Initial energy of each node	2 J
Symmetric key scheme	AES-128
Public key scheme	XTR-128
Message authentication	SHA-1
Symmetric key creation	0.00008 mJ
Encryption using symmetric key	0.0013 mJ
Encryption using public key	0.0021 mJ
TX power of encrypted data using symmetric key	0.0256 mJ
RX power of decrypted data using symmetric key	0.01792 mJ
TX power of encrypted data using public key	0.31 mJ
Energy consumption for authentication	5.9 μJ

key scheme. Those results show that the hybrid key scheme's energy dissipation is close to that of the symmetric key scheme's energy consumption.

4.2. Probability of Successful Transmission. Figures 6 and 7 show the probability of successful transmission in symmetric, public, and hybrid key schemes under the TEEN protocol. In these figures, r means the probability of the basic successful transmission for each node against attackers. When r is 0.99, as Figure 6, most of the transmission trials are successful, and three key schemes give a probability more than 0.97 on average. In addition, the gap of transmission probability among the three key schemes is small. On the other hand, when r drops to 0.9, as in Figure 7, the probability gap among three schemes is increased as the amount of communication trial is increased. The gap between a hybrid and the symmetric key schemes is wider than the gap between a hybrid and the public key schemes, especially. It means if r is decreasing, it is difficult to maintain security for systems by using only the symmetric key scheme. In this case, the average of probability is 0.899, 0.866, and 0.799 for the public, a hybrid, and the symmetric key schemes, respectively. As r drops from 0.99 to 0.9, a hybrid key scheme also decrease the probability. However, it is not far from probability of the public key scheme. Finally, Figures 6 and 7 show the fluctuating point with the network lifetime, because it generates different cluster topology every round.

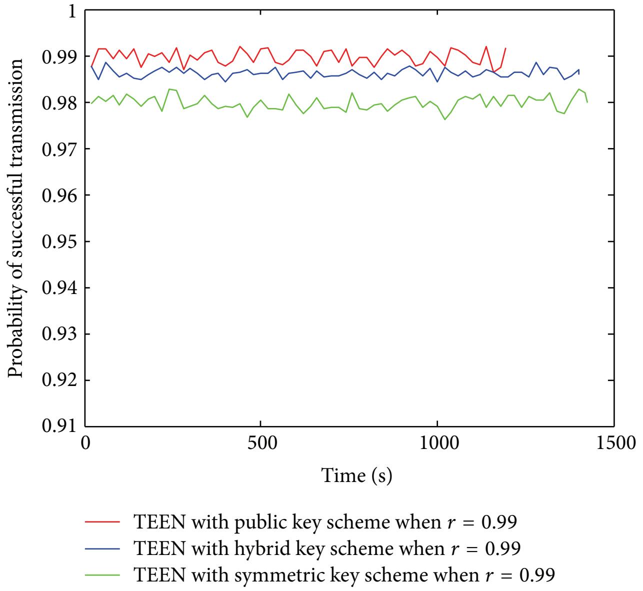


Figure 6: Probability of successful transmission as a function of time when r is 0.99.

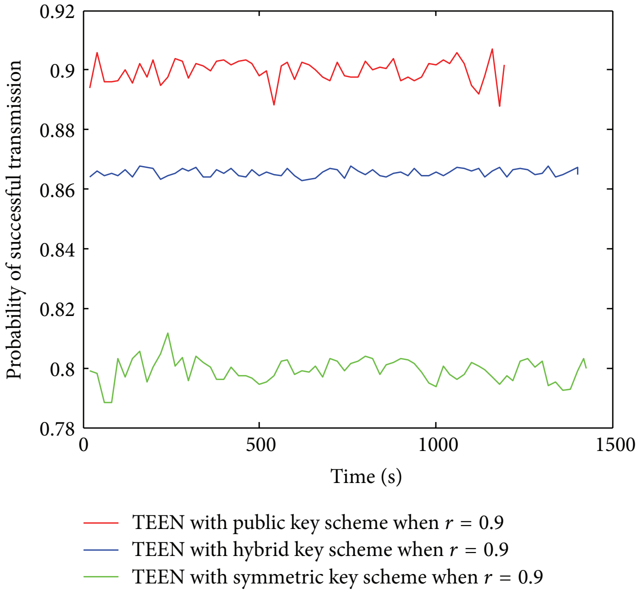


Figure 7: Probability of successful transmission as a function of time when r is 0.9.

5. Security Analysis

In this section, we analyze the security of a hybrid key scheme for the TEEN protocol. At the first part of this section, we analyze the security under several attacks which can happen for the TEEN protocol. The second part explains the security strength based on the key size.

5.1. Security Analysis. Hybrid key scheme is able to protect against typical attacks on wireless sensor networks. Various security vulnerabilities on WSNs are discussed in several literatures [5, 8]. Most attacks against WSNs routing protocols can be divided into one of the following categories: Sybil, manipulating routing information, and Hello flooding attacks. Hence, we discuss how a hybrid key can protect from various security vulnerabilities as follows.

5.1.1. Protecting from the Sybil Attack. The Sybil attack is an attack which a node sends multiple identities to other nodes. Authentication is used to verify the identity of the sender of a communication. So a node is hard to pretend to be another node. That is, when a node i transmits a sensed data to node j, it can send a MAC (Message Authentication Code) computed using the shared key $S^{r_i \times r_j}$ between nodes i and j. Since the shared key is only known by two nodes due to their random numbers r_i and r_j, nobody can pretend to be node i or j. Therefore, a hybrid key scheme is able to protect from the Sybil attack.

5.1.2. Protecting from the Manipulating Routing Information Attack. In this protocol, the routing information is distributed by each CH for each cluster. When a CH_j sends a TDMA schedule to cluster members, it appends ID_j to each message and creates the hashed value for the message. Only cluster members can check the validation of the message due to the hashed value, thus an adversary is difficult to send inexact routing information.

5.1.3. Protecting from the Hello Flood Attack. In TEEN protocol, non-CH nodes decide the cluster to which they want to belong based on signal strength. It means that a powerful advertisement can make the malicious attacker be the CH. However, identification of each node is used to verify identities of neighbor. Thus a hybrid key scheme is able to prevent from the Hello flood attack.

5.2. Security Strength Analysis. In this part, we explain the security strength of hybrid key scheme by calculating the amount of time an attacker may need in order to break the key scheme. In general, MIPS (Million Instructions Per Second) is widely accepted as a unit to approximate computation that can be performed [15].

The security strength analysis is based on the key size of the key schemes. The shorter the key size is, the more vulnerable the encrypted data becomes to exhaustive key brute-force attack. In this hybrid key scheme, as changing the role of CHs at every 20 seconds, the symmetric key which used communication for CH-to-node is also updated at every 20 seconds for each cluster. So an attacker should finish the calculation within 20 seconds to break encrypted data. In fact, it would take about 5,300,000 years using a PC with 3 GHz to break an 80-bit symmetric cipher [16]. Assume that an attacker uses the Cray Jaguar which is wellknown as the fastest operational supercomputer with a sustained processing rate of 1.759 PFLOPS (Peta FLoating point Operations Per Second) in November 2009 [17]. To break a 128-bit encrypted data, it may take over than 2.4×10^{13} years. Using a 128-bit key with AES indicates that the encrypted data can be protected.

6. Conclusion

In WSNs, each sensor node has limited resources in many hostile and tactical scenarios and important commercial applications. So TEEN routing protocol is proposed for time critical applications and energy consumption. However, most routing protocols including TEEN protocol for WSNs do not include security at the designing stage. Hence, attackers can easily attack by exploiting vulnerabilities. Also, because a wireless channel is open to everyone, it can provide easy way for attackers to break into WSNs. Therefore, WSNs demand security to protect from attackers in the design of WSN protocols.

As security is becoming a major concern for WSNs protocol because of the wide security-critical applications of WSNs, several countermeasures have been proposed, such as authentication, identitying verification, and bidirectional link verification [5]. Alternatively, key establishment is the first step to establish a security, since all encryption-decryption and authentication methods use keys. Generally, two types of key schemes are used in cryptography: symmetric and public key schemes. The former is computationally inexpensive and needs small time for calculation, though it is difficult to manage the symmetric keys. On the other hand, even if the latter is much more expensive, it gives easier key management and resilient to node compromise than the former.

In this paper, we propose a hybrid key scheme that uses both symmetric and public key schemes in order to take the advantage of the rapid calculation times of the symmetric key scheme and flexible key management of the public key scheme, adapted to the TEEN protocol for WSNs to provide secure data transmission. Two key schemes are used depending on the types of communication for the TEEN protocol. Concretely, the symmetric key scheme is applied to intracluster communication, and the public key scheme is used for intercluster communication. Also, to provide the assurance of the identities among communication nodes, we make hashed value generated from the hash function. In this way, our scheme decreases about 8% compared with network lifetime of the TEEN protocol and about 4% compared with that of the TEEN protocol with symmetric key scheme. On the other hand, when r drops from 0.99 to 0.9, the probability of a hybrid key scheme is decreased; however, a hybrid key scheme provides better probability of successful transmission than that of the symmetric key scheme.

Acknowledgments

This work was partially supported by Leading Foreign Research Institute Recruitment Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (MEST) (K20902001632-10E0100-06010) and by the World-Class University Program funded by the Ministry of Education, Science, and Technology (MEST) through the National Research Foundation of Korea (R31-10026).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] M. Yu, K. K. Leung, and A. Malvankar, "A dynamic clustering and energy efficient routing technique for sensor networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 8, pp. 3069–3079, 2007.

[3] M. Arati and P. A. Dharma, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks," in *Proceedings*

of the 15th International Parallel & Distributed Processing Symposium, pp. 2009–2015, San Francisco, Calif, USA, April 2001.

[4] E. Shi and A. Perrig, “Designing secure sensor networks,” *IEEE Wireless Communications Magazine*, vol. 11, no. 6, pp. 38–43, 2004.

[5] C. Karlof and D. Wagner, “Secure routing in wireless sensor networks: attacks and countermeasures,” in *IEEE Workshop on Sensor Network Protocols and Applications*, pp. 113–127, May 2003.

[6] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, “A pairwise key predistribution scheme for wireless sensor networks,” *ACM Transactions on Information and System Security*, vol. 8, no. 2, pp. 228–258, 2005.

[7] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 41–47, November 2002.

[8] D. W. Carman, P. S. Kruus, and B. J. Matt, “Constrainsts and approaches for distributed sensor network security,” Tech. Rep., NAI Laboratories, 2000.

[9] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS ’00)*, January 2000.

[10] M. Arati and P. A. Dharma, “APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks,” in *Proceedings of the 16th International Parallel and Distributed Processing Symposium*, pp. 195–202, 2002.

[11] S. D. Muruganathan and A. O. Fapojuwo, “A hybrid routing protocol for wireless sensor networks based on a two-level clustering hierarchy with enhanced energy efficiency,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’08)*, pp. 2051–2056, April 2008.

[12] S. Nikoletseas, I. Chatzigiannakis, H. Euthimiou, A. Kinalis, A. Antoniou, and G. Mylonas, “Energy efficient protocols for sensing multiple events in smart dust networks,” in *Proceedings of the 37th Annual Simulation Symposium, ANSS-37 2004*, pp. 15–24, April 2004.

[13] C. Kavitha and K. V. Viswanatha, “A hybrid reliable routing technique (HRR) for wireless sensor network,” *International Journal of Computer Science and Network Security*, vol. 9, no. 3, pp. 35–39, 2009.

[14] The Network Simulator—ns-2, http://www.isi.edu/nsnam/ns/.

[15] P. Prasithsangaree and P. Krishnamurthy, “Analysis of tradeoffs between security strength and energy savings in security protocols for WLANs,” in *Proceedings of the Vehicular Technology Conference*, vol. 7, pp. 5219–5223, September 2004.

[16] A. K. Lenstra and E. R. Verheul, “Selecting cryptographic key sizes,” *Journal of Cryptology*, vol. 14, no. 4, pp. 255–293, 2001.

[17] TOP500 Supercomputing sites, http://www.top500.org/.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 802526, 16 pages
http://dx.doi.org/10.1155/2013/802526

Review Article

New Trends in Secure Routing Protocols for Wireless Sensor Networks

Aly Mohamed El-Semary[1,2] **and Mohamed Mostafa Abdel-Azim**[1,3]

[1] *College of Computer Science and Engineering, Taibah University, Al-Madina Al-Munawwarah, Saudi Arabia*
[2] *Faculty of Engineering, Al-Azhar University, Cairo, Egypt*
[3] *Faculty of Industrial Education, Beni-Suef University, Beni-Suef, Egypt*

Correspondence should be addressed to Mohamed Mostafa A. Azim; mmazim@ieee.org

Received 21 February 2013; Accepted 15 April 2013

Academic Editor: S. Khan

Recently, Wireless Sensor Networks (WSNs) have been deployed into a variety of applications including homeland security, military systems, and health care. Sensor nodes deployed in such networks are subject to several attacks such as sinkhole and select forwarding, wormhole, Hello flood, and replication attacks. Therefore, developing secure and energy-efficient routing protocols to protect WSNs against these attacks while efficiently utilizing the energy of the deployed nodes has become imperative. Several routing protocols have been proposed in the literature for WSNs. Most of these protocols assume static nodes and sinks to collect data from network fields. However, they may be highly movable, and recent advances show that mobile sensors in WSNs have a promising performance. Therefore, this paper surveys the state of the art on routing protocols related to WSNs and presents the security issues or problems associated with the current protocols as well as discusses the future trends and open research issues on secure routing protocols of WSNs.

1. Introduction

Wireless Sensor Networks (WSNs) are emerging networking technologies for low-cost, unattended monitoring of a wide range of environments. WSNs received a great importance in the current decade due to the recent development of the IEEE 802.15.4 standard [1] for the physical and MAC layers and the forthcoming ZigBee standard [2] for the network and application layers. Moreover, the latest delivery of the IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) standard [3] enables us to communicate with devices across the Internet directly without having to go through a translation step such as a ZigBee-to-IP translation.

A WSN typically consists of tens, hundreds, or even thousands of low-cost, low-power, and multifunctional sensor nodes that are deployed in a region of interest [4] as shown in Figure 1. These tiny sensor nodes are equipped with sensing functionalities, embedded microprocessors, radio transceivers, and small memory. They communicate over a short distance wirelessly and cooperate to accomplish a common task, such as environmental monitoring.

In this paper, we survey the state of the art on the routing protocols used in WSNs. We demonstrate different security issues to be taken into consideration when deploying WSNs indicating their security requirements for such networks and demonstrating different types of attacks that may face. We also summarize the state-of the art secure routing protocols proposed for WSNs. Finally, we present the future directions and open research issues or problems in the field of routing protocols of WSNs.

The remainder of this paper is organized as follows. Section 2 gives an overview of the applications of wireless sensor networks. Section 3 presents a summary of the state of the art routing protocols used in wireless sensor networks. Section 4 demonstrates security issues in wireless sensor networks. Section 5 presents the up to date secure routing protocols employed in WSNs. Next, Section 6 discusses the recent research issues related to routing protocols of

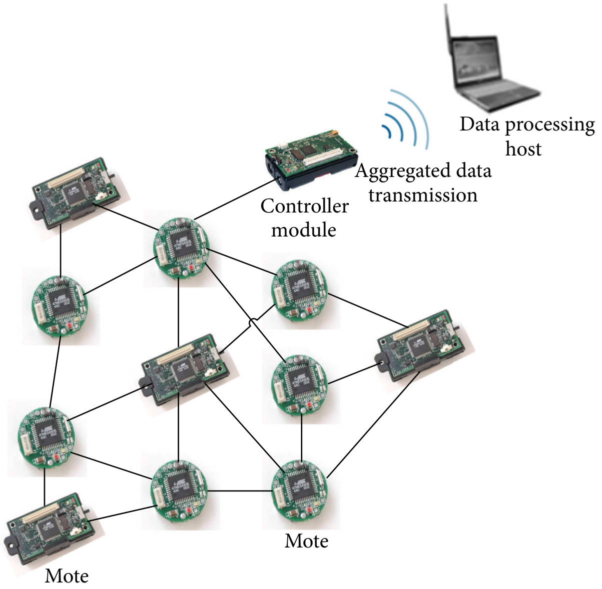


Figure 1: Illustration of a wireless sensor network [4].

WSNs. Finally, the paper concludes with the future research directions in Section 7.

2. Applications of Wireless Sensor Networks

WSNs are currently being employed in a variety of domains ranging from commercial, industrial, environmental, and healthcare to military applications to monitor data that would be difficult or expensive to capture using wired sensors [5, 6]. Based on these fields, a variety of applications have been presented in the literature including aircraft monitoring [7], ecological habit monitoring [8], and geological monitoring [9]. Wireless biomedical sensor networks (WBSNs) [10] are another application domain for WSNs that is characterized by the necessary low error rates compared to traditional WSNs.

Another trend in the WSNs applications is the Vision-Enabled Wireless Sensor Networks [11] which is a new application platform for Wireless Image Sensor Network. Application examples of this category include Event Detection, Multimodal Node Localization, Collaborative Self-Localization, Traffic Monitoring, and Target Tracking.

3. Wireless Sensor Networks Routing Protocols

Due to the natural limitations imposed on sensor nodes, several network layer protocols have been proposed to utilize sensor's energy to prolong the life time of deployed wireless sensor networks (WSNs). These protocols can be generally classified into four main categories: data-centric, hierarchical, location-based, and Quality of Service- (QoS-) based routing protocols. Extensive surveys on routing protocols can be found in [12–18].

3.1. Data-Centric Protocols. Data-centric routing protocols are negotiation-based and application-specific protocols. In data-centric routing, the sink node forwards queries to selected regions and waits for data replayed for the sensors in the specified regions. While data is being requested through queries, attribute-based naming should be utilized to identify the characteristics of required data. Several data-centric protocols have been proposed in the literature including Flooding and Gossiping [19], SPIN [20], DD [21], and Rumor [22].

Flooding and Gossiping are two classical algorithms to convey data in sensor networks without any routing protocols and topology maintenance. In Flooding algorithm, when a sensor receives a data packet, it broadcasts the data packets to all neighbors within its transmission range. Each sensor receiving the packet repeats the same process until the packet reaches its either destination or time to live. Even though Flooding algorithm is straightforward to implement, it has several disadvantages including *implosion, overlap, and resource blindness*. Implosion occurs when a node receives the same packet from more than one node. Gossiping algorithm overcomes the shortcoming of implosion by just making each receiving node forwards the data packet to a randomly chosen neighbor which in turn selects another random neighbor to forward the packet to and so on.

Sensor Protocols for Information via Negotiation (SPIN) is a family of adaptive routing protocols. SPIN is designed to improve classical mechanisms to overcome the problems of implosion and overlap. It gives a high-level name (called metadata) to completely describe the collected data, and also it uses three types of messages: advertisement (ADV), request (REQ), and DATA to convey data messages throughout the network. In SPIN protocols, when a node has a data to convey, it first sends an ADV message attached with a metadata to its neighbors for negotiation. Next, each node receiving the ADV message replays with a REQ message if it is interested in the data. Then, when the node that has the data receives the REQ messages, it starts to send the DATA message to each node sending a REQ. Finally, once a node receives the DATA message, it repeats the same sequence of steps to convey the DATA messages to all interested nodes all over the network. SPIN has the advantage that it provides more energy savings over the classical protocols. This is achieved through using metadata negotiation which almost reduces the redundant data halves. On the other hand, the data advertisement mechanism of SPIN cannot guarantee delivery of data. The implosion problem still exists during the REQ phase and the solution for the collision is not considered.

Directed Diffusion (DD) is a routing protocol for distributed activities where a large number of limited resource nodes locally coordinate to achieve a sensing task. The DD protocol has two main features: *Interests* and *Gradients*. The *Interest* is a named data or task description to define the sensor events that an originator is interested in, and it has a list of several attribute-value pairs including type, region, rate, duration, and time stamp. The type is used to impose what type of data should be sensed, and the region defines the network part from which the data should be drawn. The rate attribute sets how often the data should be forwarded, while the duration sets how long the interest should be active, and the time stamp is used to refresh the interest. Each interest entry has several gradients, one to each neighbor. Each gradient also has a set of attributes including a node identity to which the data will be forwarded, data rate that tells how often the data will be forwarded, and the duration field which sets how long the gradient will be active. The DD protocol is completed by diffusing the interest, setting up gradients, and sending data and path reinforcement.

Rumor routing algorithm is a variation of DD, and it is characterized by events, queries, and agents. An event is a phenomenon occurring in a fixed region of the network. A query is a request for information or an order to collect data, while an agent is a long-lived packet used to create paths leading to events. Each node on the path contains both a list of its neighbors and an event table in which a row has an event name, number of hops to the event, and the next node in the path toward the event. In the rumor routing, any node can generate a query and then send the query in a random walk to find the path. The query keeps going on until it finds the path or its time to live (TTL) expires. If the node originated, the query discovers that the query is dead; without finding a path to the event, it can retransmit, stop, or flood the query.

3.2. Hierarchical Routing Protocols. Hierarchical routing protocols often group sensor nodes into clusters that form a hierarchy. Several hierarchical protocols have been introduced in the literature including LEACH [23], TEEN [24], and PEGASIS [25].

Low Energy Adaptive Clustering Hierarchy (LEACH) is a self-organizing, adaptive clustering protocol that utilizes randomization to balance the energy load among the deployed nodes in the network. LEACH assumes that all nodes start with uniform energy distribution, and all nodes can communicate directly with the base station. It works in two rounds, set-up and steady-state. The set-up round is responsible for forming clusters and cluster heads. During this round, sensors vote for themselves to be cluster heads at any given time with certain probability based on their energy. Next, the cluster head advertises their status to other sensors in the network. Accordingly, each sensor node decides the cluster that it wants to belong to by selecting the cluster head that needs the least amount of communication energy. Finally, once all the nodes are arranged into clusters, each cluster head creates a schedule for the nodes in its cluster to avoid collision. This results in the network structure that consists of a base station or sink, cluster heads that communicate with the sink, and sensor nodes each is recognized by a cluster and communicated with its cluster head that is a single hop to the sink.

The steady-state phase is concerned with transferring the data from the sensors in the network to the base station or sink node. During this phase, the cluster head collects the data from the sensors in its cluster. Once the data is collected from all nodes in the cluster, the cluster head locally aggregates the data in some way based on the application to remove the unreliable data. Next, the cluster head transmits the collected data to the base station. LEACH uses the local processing to reduce global communication and also randomizes the rotation of cluster heads. Therefore, it prolongs the network's life time. On the other hand, it is not applicable to large networks and time-critical applications. In addition, the idea of dynamic clustering brings extra overhead. Furthermore, the cluster heads send data to the sink through high power link which make these clusters consume their energy faster.

Threshold-Sensitive Energy-Efficient Protocols (TEEN) are designed for time-critical applications in which a sudden change in the measured attribute (e.g., temperature) may happen. The TEEN networks employ multilevel clustering mechanism to avoid the transmission over high power link implemented in LEACH. In TEEN, once the clusters and cluster heads are formed, each cluster head broadcasts two main parameters: hard threshold (HT) and soft threshold (ST) to the sensor nodes in its clusters. Each sensor in the network has a variable called SV to store the sensed value. Also, each sensor node continuously measures the environment and if the measured value is above its HT for the first time, the node stores the value into SV, turns on its transmitter, and sends the measured value to its cluster head which in turn forwards it to another cluster head until it reaches the sink node. The node will next transmit the measured value in the current cluster period if two conditions are met: (1) the current measured value is greater than HT and (2) the current measured value varied from the SV by the value of ST. Using this technique, TEEN reduces more network traffic and extends the life time of the LEACH networks into two things: (1) it reduces the large energy consumed by transmission over high power link through utilizing multilevel clustering algorithm and (2) it reduces the network traffic through implementing the HT and ST.

Power Efficient Gathering in Sensor Information Systems (PEGASIS) implements a chain-based algorithm to eliminate the overhead resulted from dynamic formation of cluster in LEACH and TEEN. PEGASIS assumes that all nodes know the overall topology and implements greedy algorithm to form data chain. To form the data chain, a downstream node chooses the closest node as an upstream node from the unchained nodes. Once the data chain is formed, the last node of the chain is called the chain leader which begins to send data to the base station. Any node on the chain receives the data from its downstream, aggregates the data with its own in some way, and then sends the aggregated data to its upstream node. This process is repeated until the data reaches the chain leader which in turn sends it to the base station. The data chain is constructed once and followed by several rounds of data communication, but if a node on the chain is dead, the chain is reconstructed again in the same manner to avoid the dead node. Even though PEGASIS overcomes the overhead resulted from dynamic cluster formulation by constricting data chains, it has several drawbacks: (1) it assumes that each node can communicate directly with the BS, (2) the chain leader can become a bottleneck, and (3) it uses greedy algorithm which is locally efficient but globally inefficient.

3.3. Location-Based Routing Protocols. Location-based routing protocols are a family of routing protocols in which each deployed sensor node should know its local location information by some means, for instance, GPS [26]. In addition, it also may require that each sensor node is aware of its remaining amount of energy. It uses this information as forwarding metrics during the forwarding of data packets from a source node to a distention node. This family of protocols include GEAR [27], Min-Hop [28], MAP [29], and PEW [30].

Geographic and Energy-Aware Routing (GEAR) is designed for routing queries to specific regions. It assumes that all sensor nodes are aware of their remaining energy and their locations by some means such as GPS. In addition, each node should know the location and remaining energy of it neighbors simply by Hello flood messages. GEAR utilizes this information to construct a heuristic function that avoids energy holes and chooses sensors to route a packet toward the target region. This forwarding process is repeated until the packet is disseminated inside the target region.

Minimum hop (Min-Hop) routing protocol forwards packets from a source node to the sink node through an optimal path. The optimal path is the route that has the shortest path to the sink. The path is represented as the number of hops to the sink. Therefore, the source node chooses the node of the next hop as the one that has the shortest path or the minimum number of hops to the sink. If several nodes have the same number of hops to the sink, the one with the maximum energy is considered. This forwarding process is continued until the packet reaches either the sink node or its predefined maximum of hops to avoid packet loop. The Min-Hop works in two phases: initialization phase and routing phase. The initialization phase is concerned with building the routing tables of the network nodes, while the routing phase starts after initialization phase, and it is responsible for forwarding data packets through the network until they reach the sink node. Although the Min-Hop improves the average energy consumption in the network by using the shortest paths, it overutilizes the nodes along the shortest paths. This results in increasing energy gaps or holes and decreasing network life time.

Maximum Available Power (MAP) is an enhanced version of the Min-Hop to lengthen the network life time. The MAP distributes network load by choosing the neighbor node with the maximum energy, but this leads to significantly longer paths. Both Min-Hop and MAP forward data packets based on local view of neighbors and thus may result in energy holes that lead to decreasing the network life time. In other words, selected paths may include one or more nodes with very low energy.

Path Energy Weight (PEW) overcomes this shortcoming by deploying a global view mechanism. Its main idea is to globally map energy levels of all nodes along the communication path into a single parameter. This parameter reflects how uniform the energy distribution along the path is. It gives a better weight to a path with balanced energy level over a path with unbalanced energy level.

3.4. QoS-Based Routing Protocols. Wireless Multimedia Sensor Networks (WMSNs) [31, 32] are a new powerful class of sensor-based distributed intelligent systems capable of ubiquitously retrieving multimedia information derived by the recent technological advancements in microelectromechanical systems. WMSNs consist of sensor nodes equipped with audio and visual information collection modules that have the ability to retrieve multimedia data, store or process data in real time, correlate and fuse multimedia data originated from heterogeneous sources, and wirelessly transmit collected data to the desired destinations.

In the QoS-based routing protocols, routing process of selecting the next forwarding node is performed based on Quality of Service (QoS) parameters such as bounded latency or delay, bandwidth, jitter, and reliability. However, satisfying the stringent QoS requirements of multimedia transmission in a resource-constrained sensor network environment places new challenges to routing. Hence, optimal energy and application-specific QoS-aware routing for WMSNs have gained considerable research attention recently. The authors in [33] claim that classic multipath routing approaches are vulnerable to black holes attacks, mainly due to their deterministic nature. They also proposed mechanisms that generate randomized multipath routes. Besides routes randomization, the generated routes are also highly dispersive and energy efficient, making them quite capable of circumventing black holes.

The authors in [34] focus on the selective jamming attacks in WSNs. To face these attacks, they developed three schemes that prevent real-time packet classification by combining cryptographic primitives with physical layer attributes.

In [35], the authors propose a secure node disjoint multipath routing protocol in which the data packets are transmitted in a secure manner by using the digital signature cryptosystem. Their proposed protocol can improve the packet delivery and reduce the end-to-end delay.

The authors in [36] formulated the secret-sharing-based multipath routing problem as an optimization problem aiming at maximizing both network security and life time, subject to the energy constraints. They also propose a three-phase disjoint routing scheme called the Security and Energy-efficient Disjoint Route (SEDR). Based on the secret-sharing algorithm, the SEDR scheme can depressively and randomly deliver shares all over the network in the first two phases and then transmits these shares to the sink node. The proposed scheme improves the network security under both scenarios of single and multiple black holes without reducing the network's life time.

4. Security Issues in Wireless Sensor Networks

Due to the limited resources of the sensor nodes in wireless sensor networks, most of the traditional security mechanisms employed in traditional wireless networks are not applicable for wireless sensor networks (WSNs). Thus, applying a security mechanism into a WSN is a challenging task. The rest of this section introduces the main security issues associated with WSNs. These main issues are security requirements, attacks in WSNs, and key managements [37, 38].

4.1. Security Requirement in WSNs. To accomplish security in WSNs, the basic security requirements or objectives should be applied to face passive attacks, active attacks, and Denial-of-service (DoS) attacks [39, 40]. Passive attacks are silent in nature and do not affect the normal network operations;

therefore they are very difficult to detect. However, passive attacks lay down a foundation for later launching an active attack. Integrity and availability are compromised by active attacks, while the confidentiality of end-users' traffic is compromised by passive attacks. The confidentiality is the process of hiding messages so that unauthorized entity cannot reveal them. The integrity is the process of ensuring that messages are not altered by unauthorized entity. The availability is the process of ensuring that network services are available for authorized entities when they are needed. In wireless networks, these requirements are independent of application due to the data aggregation at the intermediate nodes. However, some systems may satisfy some of the requirements based on the application [41]. The DoS attack will be discussed in details in the following subsection.

4.2. Attacks on Routing Protocols of WSNs. Wireless sensor networks are subject to several types of attacks which can be categorized into attacks on information in transit, node replication attack, Denial-of-service, and routing attacks. Since the routing protocols are the scope of this paper, the first three categories are discussed briefly hereafter, and the rest of section is devoted for routing attacks.

Attacks on information in transit are the most common attacks against WSNs in which the information in transit is vulnerable to eavesdropping, modification, injection, interruption, and traffic analysis. Most of these attacks can be thwarted by implementing confidentiality, authentication, and integrity. In a node replication attack, an attacker can insert a new node into a network which has been cloned from an existing node. This new node can act exactly like the old node, or it can have some extra behavior, such as transmitting interested information directly to the attacker. The Denial-of-Service (DoS) attacks can take place at different layers. At the physical layer, one or more attackers continuously transmit a radio signal that interferes with the radio frequency used by the sensors. This jamming can render the network sensors to be ineffective. Also, DoS attack can occur at the data link layer by violating the communication protocol. For example, a malicious node continuously transmits messages to generate collision or to deplete the power of the target nodes [42]. DoS attacks at network layer involve attacking the routing protocols which is discussed in the rest of this section.

A large number of the routing protocols of wireless sensor networks are designed without security in mind. Consequently, these protocols are subject to several attacks associated with the network layer of ad hoc networks. Most of these attacks fall into one of the following categories: acknowledgment spoofing attack [42], false routing information attack [43], wormhole attack [44], selective forwarding attack [45], sinkhole attack [46], Hello flood attack [46], and Sybil attack [47].

In the acknowledgement spoofing attack, an adversary may forge on acknowledgement aiming to influence that a weak link is strong or a dead node is alive. As a result, a weak link may be chosen for routing, and hence packets delivered through that link may be lost or corrupted. An attacker using acknowledgment spoofing attack can effectively mount a selective forwarding attack by encouraging the target node to send out packets through those weak links.

In the false routing information attack, an attacker can reshape the whole network by conveying false routing information to the other nodes in the network. As a result, the attacker can create routing loops, attract or repel network traffic from specific nodes, extend or shorten source routes, generate fake error messages, cause network partitions, increase end-to-end latency, and reduce the network life time.

In the wormhole attack, two adversaries cooperated to tunnel messages received from sensors at a network location over a low latency link and then replay them to sensors at a different location. The wormhole attack is dangerous for WSN routing protocols since attackers can achieve it without compromising any sensor node in the network; even all of the sensor nodes in the network utilize effective authentication and confidentiality mechanism.

In selective forwarding, malicious nodes may not forward specific messages and simply drop them. A simple aspect of the selective forwarding called sometimes grey-hole attack is when a malicious node acts like a black hole by refusing to forward every observable packet. The selective forwarding attack is most effective when the attacker's node is able to include himself on the forwarding path of target messages.

In the sinkhole attack, one or more malicious nodes prevent the base station from obtaining complete and correct sensing data, thus forming a serious threat to higher-layer applications. The sinkhole attack is achieved by making a compromised node look attractive to its neighbor nodes with respect to the routing metrics. Consequently, the attacker manages to draw as much traffic as possible that is designated to the base station. By involving himself in the routing process, the attacker is then able to launch more severe attacks such as selective forwarding, modifying or dropping the received packets.

In the Hello flood attack, an adversary exploits the behavior of most routing protocols requiring each node to broadcast a Hello message to discover its neighbor and automatically create a network. Nodes receiving this message mark the sending node as their parent. As a result, each of these nodes will forward the packets to its parent. In the Hello flood attack, an attacker can use a wireless device with a large enough transmission power for compromising all nodes in the network that this device is its parent. For example, if an attacker broadcasts a Hello message with a large enough transmission power, the malicious node will convince each node in the network that the adversary is its neighbor and marks the adversary as its parent. This makes all nodes send their messages to the adversary, but because they are far away from this adversary, they send them to oblivion.

In the Sybil attack, a malicious node unlawfully claims several identities to represent a large number of other sensor nodes in the network. In this attack, each of these identities is called a Sybil node which can be obtained be an attacker in one of two ways. In the first way, the attacker can simply generate a new Sybil identity from the address space to each Sybil node. For example, if a node is identified by a 16-bit integer, the attacker can simply assign a random 16-bit integer for each Sybil node. In the second way, the attacker can steal or spoof a legitimate identity for each Sybil node. The way is most likely used when the address space is intentionally limited to prevent attackers from injecting any new identity. Once each of the Sybil nodes is assigned an identity, it can communicate with legitimate nodes directly or indirectly through malicious nodes. Consequently, the Sybil attack imposes a significant impact on the routing process of the underlying network.

4.3. Key Management in WSNs. The wireless sensor networks (WSNs) have a set of security requirements that must be achieved to protect the networks against most of the associated attacks. These security requirements are confidentiality, integrity, availability, authentication, and refreshment. To provide these requirements, a key management mechanism suitable for WSN must be implemented. The key management in WSN is a set of key distribution mechanisms; each mechanism is responsible for establishing cryptographic key or key material among all sensors nodes in the network. In addition, it is concerned with revoking and refreshing keys. A good key distribution mechanism should have the following features: scalability, efficiency, connectivity, and resilience

The scalability means that the key distribution mechanism should support large networks and be flexible against substantial increase in network size even after deployment. Efficient mechanism should consider sensors limitations such as storage, processing, and communication. Key connectivity is the probability that two or more nodes share the same key. Enough key connectivity must be provided for a WSN in order to perform its task. Resilience means that the algorithm is resisting against node capture. For example, compromising the security credentials stored on a sensor node should not reveal information about the secrecy of any other links in the WSN. These features conflict with each other, and thus a tradeoff should be considered. Accordingly, researchers developed several key distribution mechanisms including [48–52].

5. Secure Routing Protocols

Even though the above mentioned routing protocols utilize the limited capabilities of sensor nodes, they have not been designed with a security goal in mind. Consequently, they are not applicable into adversarial environment, such as military systems and disaster relief, due to their susceptibility to a great number of attacks against routing protocols. These attacks include selective forwarding, sinkhole, wormhole, and Sybil described in Section 4. To protect WSNs against these routing attacks, a number of secure routing protocols have been proposed in the literature including SPINS [53], SSPIN [54], ITSRP [55], COOL [56], EESRP [57], DSSRP [58], TTSS [59], SRPMND [60], and STAPLE [61].

SPINS enabled security features through two security protocols: SNEP and μTESLA. SNEP provides confidentiality, integrity, authentication, and freshness, while μTESLA [62] provides authenticated broadcast. SPINS gives more attention on key management.

Secure SPIN (SSPIN) is a secure version of SPIN family discussed in Section 3. It uses Message Authentication Code (MAC) to provide integrity and packet correctness, but it did not clearly specify how the confidentiality is achieved. Like SPIN, SSPIN works in three stages: ADV, REQ, and DATA. When a node has new data, it creates ADV message with its MAC and then broadcasts the ADV to its neighbors. Once a node obtains the ADV, it verifies the ADV through its MAC. If it is a valid ADV, the receiver sends the REQ message to a source which in turn will verify the REQ through the associated MAC. If the REQ is valid, the DATA stage starts. The source generates the MAC, attaches it to the data packet, and then sends the packet to the destination. The authors prove that SSPIN is secure if MAC scheme is secure against existential forgery attack.

ITSRP (Intrusion-Tolerant Secure Routing Protocol) applies mechanisms for authenticated key exchange and energy factor as well as resistance to some types of attacks such as sinkhole and wormhole attacks. The authors of ITSRP developed Distributed key management scheme running at the sink. It is responsible for distributed and session key initialization. ITSRP uses cryptography to provide authentication and data secrecy.

COmpromised nOde Locator (COOL) is an authentication scheme proposed by Zhang et al. [56]; the main idea of this protocol is to detect and locate compromised nodes once they misbehave in the network based on the observation of well-behaved sensor nodes.

Energy-Efficient Secure Routing Protocol (EESRP) is designed to provide both security and energy efficiency through developing two protocols: Roulette-Wheel Routing Protocol (RWRP) and Secure Routing Protocol (SRP). The RWRP is developed to forward data packets from a source to the sink node. The forwarding decision of each node is independent from other nodes (i.e., the node does not collect information from other nodes to make the decision). Thus, the node decision cannot be deceived by other nodes. SRP is interested in securing data during its traveling from a source to the sink. It implements μTESLA to authenticate packets from the sink. In addition, it used shared keys, Message Authentication Code (MAC), and time stamp to provide confidentiality, authentication and integrity, and refreshment of packets, respectively.

Dual Sink Secure Routing Protocol (DSSRP) is an enhanced version from the EESRP to prolong the network life time by using two sinks. Like the EESRP, DSSRP provides its functionality through two protocols: Next Node Selection Protocol (NNSP) and Network Protection Protocol (NPP). NNSP and NPP are a modification version of RWRP and SRP to adapt the operation for two sinks, respectively.

TTSS (The Three-Tier Security Scheme) is a secure routing protocol designed to eliminate *mobile sink replication* attacks associated with sensor networks that have mobile sinks. It differentiates between three types of nodes: *mobile sinks*, *stationary access nodes*, and *sensor nodes*. The stationary access nodes are a small number of nodes chosen randomly from the network nodes, and they are used as access points to mobile sinks in order to collect data from sensor nodes. Also, TTSS uses two separate pools of polynomial keys: the mobile polynomial pool MP of size M and the static polynomial pool SP of size S. TTSS works in two phases: static and mobile polynomial predistribution and key discovery between mobile node and stationary node. In the first phase which was executed before deployment, all mobile sinks and stationary access nodes are randomly given as k_m and one polynomial ($k_m > 1$) from MP such that the number of polynomials in each mobile sink is greater than the number of polynomial in stationary access nodes. This is to assure that a mobile sink shares with high probability a mobile polynomial with a stationary access node and to reduce the number of compromised mobile polynomials in case a stationary access node is compromised. Also, all stationary access nodes and sensor nodes randomly pick a subset of k_s from SP. In the second phase, when a sensor node establishes a direct pairwise key with a mobile sink, it has to find a stationary access node in its neighborhood that share a pairwise polynomial with the mobile sink. Using two polynomials makes the authentication of mobile sinks independent of the key distribution of sensor nodes.

SRPMND stands for Secure Routing Protocol with Malicious Node Detecting and Diagnosing for Wireless Sensor Networks. It uses μTESLA authentication protocol to protect packets sink node against the modification, forging, and replay. Also, the protocol implements an acknowledgment mechanism to detect the malicious nodes. For example, if the node did not hear an acknowledgment within a specific period of time, this means that the packet is forwarded to a malicious node. Therefore, a secure route can be created when each node on the path forwards its packet and waits for an acknowledgement until the packet reaches the sink.

STAPLE is a secure routing and aggregation protocol with low energy cost for sensor networks. It applies one-way hash chain and multipath routing to gain the security and uses hash-based message authenticated code to achieve authentication and data integrity. In addition, it uses symmetric cryptography to provide data secrecy. STAPLE works in three stages: initialization, transmission, and source authentication. The first stage is responsible for both arranging the nodes in different levels according to the minimum hop from the sink node and distributing keys. The second stage is concerned with forwarding packets from child node to parent and so on until they reach the sink. During this stage, STAPLE achieves child authentication and data integrity authentication as well as detection of false packets. The last stage authenticates the source and verifies integrity.

In addition, the authors in [63] propose a security approach that uses secret key cryptography and key management along with rekeying support. The proposed protocol provides backward secrecy, privacy, data integrity, and secure management to a wireless sensor network. It is also capable of identifying different attacks such as replay attack, DoS attack, and Sybil attack with low-resource requirements. Khan et al. [64] proposed a robust and secure routing protocol for an infrastructure-based wireless mesh (SRPM) networks. The proposed protocol performs well against a variety of multihop threats tested over a range of networks scenarios. The authors in [65] propose a cross-layer design to improve the overall performance of multihop wireless networks. The proposed architecture provides different parameters at different layers that can be utilized in protocol design phase. The authors in [66] presented cross-layer secure and resource-aware on demand routing (CSROR) protocols for hybrid WMN which is designed to ensure routing security and provide different applications specific requirements for multimedia delivery and real-time transmissions. CSROR selects an optimum route on the basis of route security taking into consideration the different cross-layer parameters.

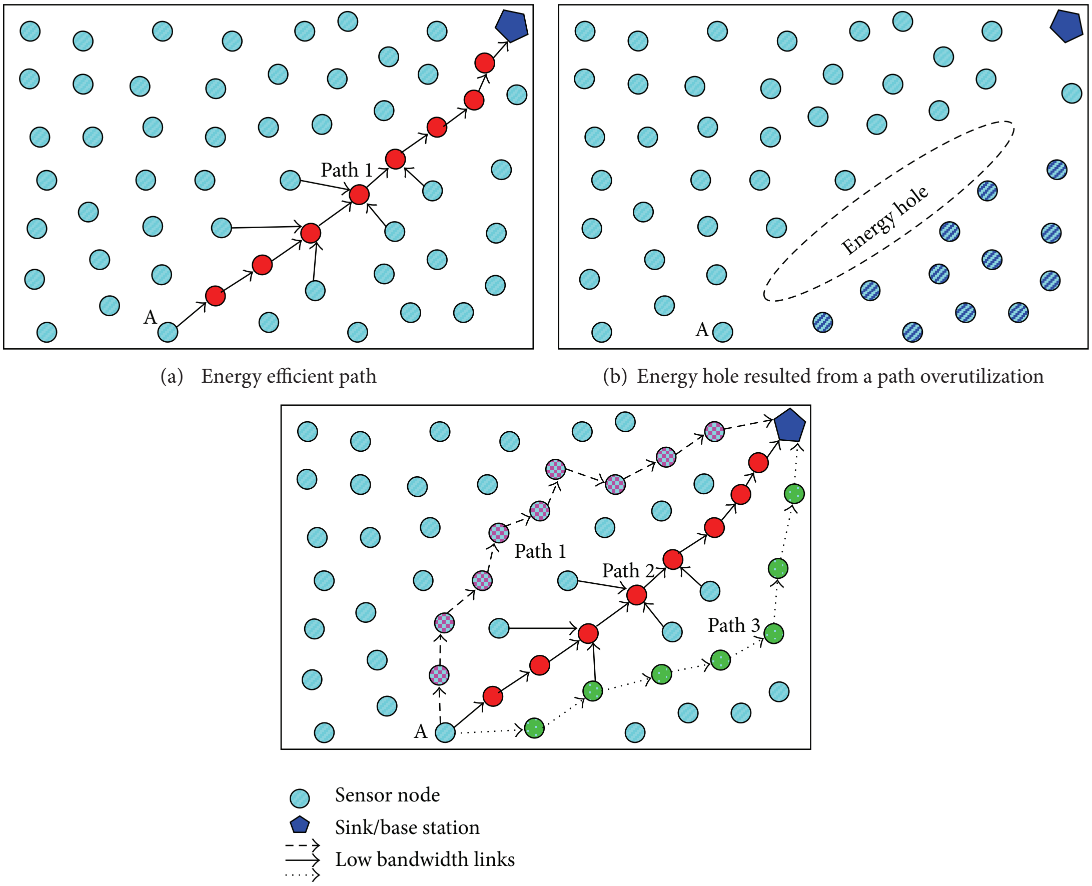
6. Recent Research Issues in WSN Routing Protocols

Due to the continual proliferation of WSN applications, especially, QoS crucial applications, the routing protocols for these networks have introduced a lot of challenges that need more study and exploration. These research challenges or issues include energy efficiency and QoS guarantee, network dynamics, multiple sources and sinks, dynamic holes bypassing, and secure routing [16]. Due to the importance of these issues, the rest of this section is devoted to discuss them for future study

6.1. Energy Efficiency and QoS Guarantee. Utilizing nodes' energy is an essential factor in designing routing protocols of WSNs to prolong the network life time of traditional WSNs. However, employing such energy-efficient routing protocols in WMSN may result in energy holes due to the large amount of data transferred in such networks. Figure 2 describes this issue in which Figure 2(a) explores the energy-efficient path from node A to the sink node. However, overutilizing this efficient path may result in energy holes along the path as shown in Figure 2(b). A possible solution to overcome the energy hole problem is to explore multipath techniques that satisfy both energy efficiency and QoS requirements between the node A and the sink node as depicted in Figure 2(c).

Multipath routing can be interpreted in two different ways. First, it can be envisioned as a multipath exploration while employing a single path randomly at a time for data transmission. The objective of the random selection of a path is to evenly distribute the energy consumption among the explored paths. In this case, multipath routing can improve the reliability and strengthen the security by avoiding the failed or compromised paths. Second, multipath routing can also be achieved by exploring multipaths and employing them for carrying the distributed data along the explored paths simultaneously. In addition to the advantages of the first scenario, the second one will reduce the end-to-end delay and facilitate the detection and prevention of selective forwarding and sinkhole attacks. However, data segmentation at the source node and aggregation at the sink node will introduce an overhead. Multipath routing providing QoS guarantee requires further studies for investigating the tradeoff between the energy efficiency and the QoS parameters.

6.2. Network Dynamics. Most of the current routing algorithms assume that network sensor nodes are static. However, they may be highly movable, and latest advances show that mobile sensors in WSNs have a promising performance [16]. Also, the recent research on data collection reveals that



(a) Energy efficient path

(b) Energy hole resulted from a path overutilization

(c) Multipath exploration

Figure 2: Energy utilization and multipath.

reporting data through implementing and leveraging sink mobility is promising for energy-efficient data gathering than reporting data through long, multi-hop, and error-prone paths to a static sink in either tree or cluster network structure [67]. Accordingly, Liu et al. [68] proposed a data reporting protocol that implements a mobile sink. Their protocol is called SinkTrail in which a mobile sink moves around the network field with relatively low velocity and keeps listening to data report packets. The mobile sink stops at some places called "trail points" for a short amount of time. Next, it broadcasts a message called "trail message" to the whole network and simultaneously listens to data report packets. Finally, it moves on to another trail point as shown in Figure 3.

The distance between any two consecutive trail points is assumed to be the same by the authors in order to facilitate the tracking of a mobile sink. The authors showed that the networks which implemented sink mobility have a great performance over networks that deployed static sink. The authors also revealed that the movement of the sink in a circular pattern outperforms each of random, angular, and linear patterns.

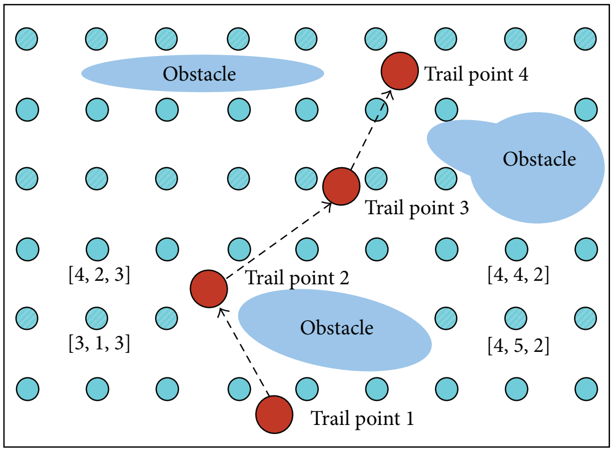
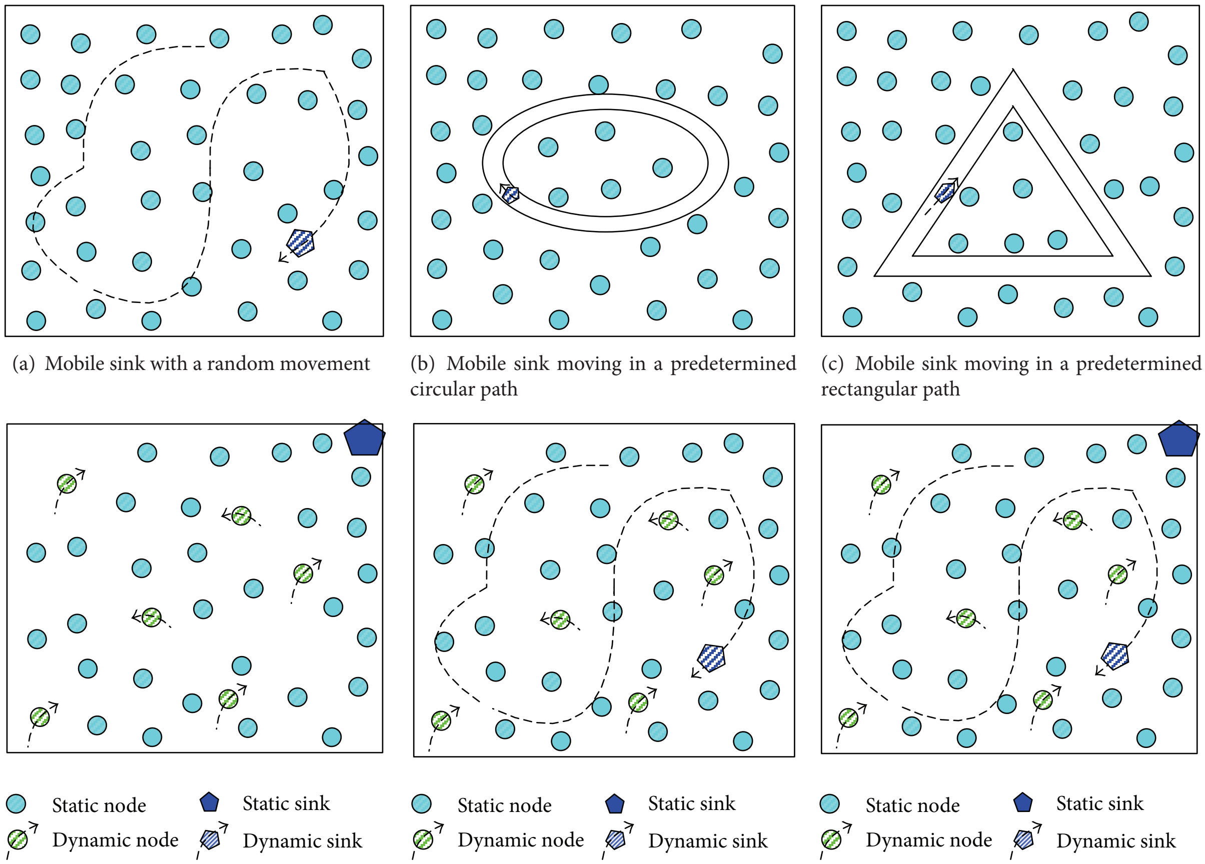


Figure 3: A network with a mobile sink [68].

Even though the recent research revealed that leveraging mobility into the WSNs has a great or promising performance, the sink mobility introduced several issues that require more consideration study. The first of these issues is that the sink mobility presented the sink replication attack in which a malicious sink claims that it is the intended one. This may reveal confidential data for the public or prevent it from reaching the authorized sink. Therefore, the researcher should consider this attack along with the other associated attacks when designing secure routing protocols implementing sink mobility. The second issue is the minimization of the



(a) Mobile sink with a random movement

(b) Mobile sink moving in a predetermined circular path

(c) Mobile sink moving in a predetermined rectangular path

(d) Static sink and a subset of dynamic nodes

(e) Mobile sink and a subset of dynamic nodes

(f) Static sink and both mobile sink and subset of dynamic nodes

Figure 4: Different trajectory patterns and techniques for network dynamics.

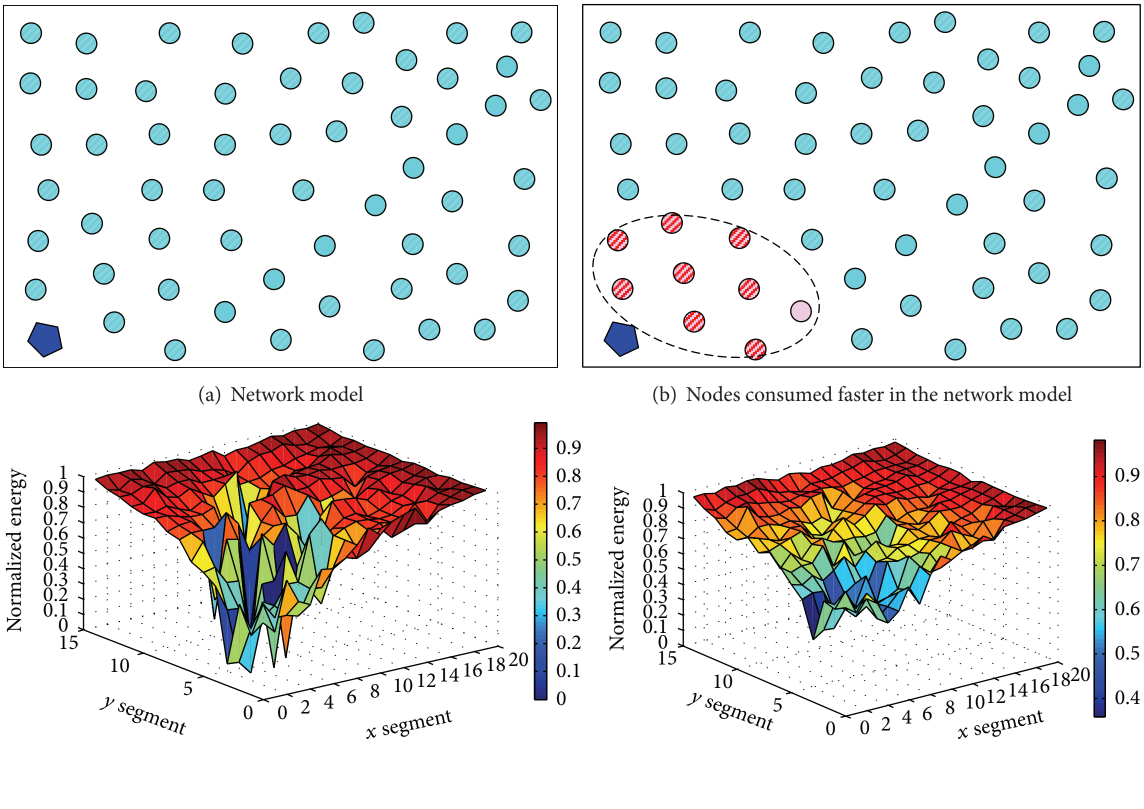
overhead resulting from the frequent updates of new location information. The third issue is what is the optimal number of mobile sinks as a function of sensor nodes deployed into the network field? The last issue, but not least, is the optimization of moving trajectory for a mobile sink since it is an NP-hard problem [69]. This issue also includes finding out the proper moving trajectory pattern such as random, circular, or triangular as seen in Figures 4(a), 4(b), and 4(c), respectively.

In addition, we need to study the performance when dynamic nodes together with a static sink or mobile sink are deployed in a network field as depicted in Figures 4(d) and 4(e), respectively. Furthermore, we need to explore the performance related to a hybrid combination of static and mobile sinks in environments with static and dynamic regular nodes as shown in Figure 4(f). Consequently, supporting mobility into routing protocols can be shown as an interesting area for future studies.

6.3. Multiple Sources and Sinks. Most of the current routing protocols forward data from a single source to a single sink. When a source node has data to share with the sink, it forwards the data to the sink node over an explored path between them. This may have several impacts on the deployed network. The network life time may be reduced due to consuming a lot of nodes' energy over long and multi-hop routes. Next, the collision may be increased when several sources send their data to the sink due to the fact that many paths from different sources to the sink have overlapping links. In addition, the nodes in the neighborhood of the sink node consume their energy faster because they are responsible for forwarding their own data along with the received data from other nodes. Consuming the energy of the neighboring nodes faster may result in a network partition in which the sink node is in one partition and the other nodes are in another partition as visualized in Figure 5 by the authors of [30].

The network partition renders the network ineffective and reduces the network life time. To clarify the point, Figure 5 visualizes the energy of the network nodes deploying either the Min-Hop or the PEW routing protocols. The network model used by any of these two protocols is depicted in Figure 5(a). It has a single static sink at (0, 0) in *xy*-coordinates, and it has also 300 nodes that are distributed randomly all over the network field. Figure 5(b) surrounds the neighbor nodes of the sink by a dashed circle. The energy of these nodes is consumed faster than the other nodes in the network as visualized in Figures 5(c) and 5(d) for the Min-Hop and PEW protocols, respectively.

Fortunately, the aforementioned problems resulted from deploying only a single sink seen in Figure 6(a) which can



(a) Network model

(b) Nodes consumed faster in the network model

(c) Energy visualization of Min-Hop found in [30]

(d) Energy visualization of PEW found in [30]

Figure 5: Remaining nodes' energy visualization in both Min-Hop and PEW networks.

be elevated by implementing multisinks. For example, the DSSRP protocol [58] implements two static sinks and reveals a promising performance over a single static sink. Also, networks supporting several sinks may be entailed to accept several events simultaneously.

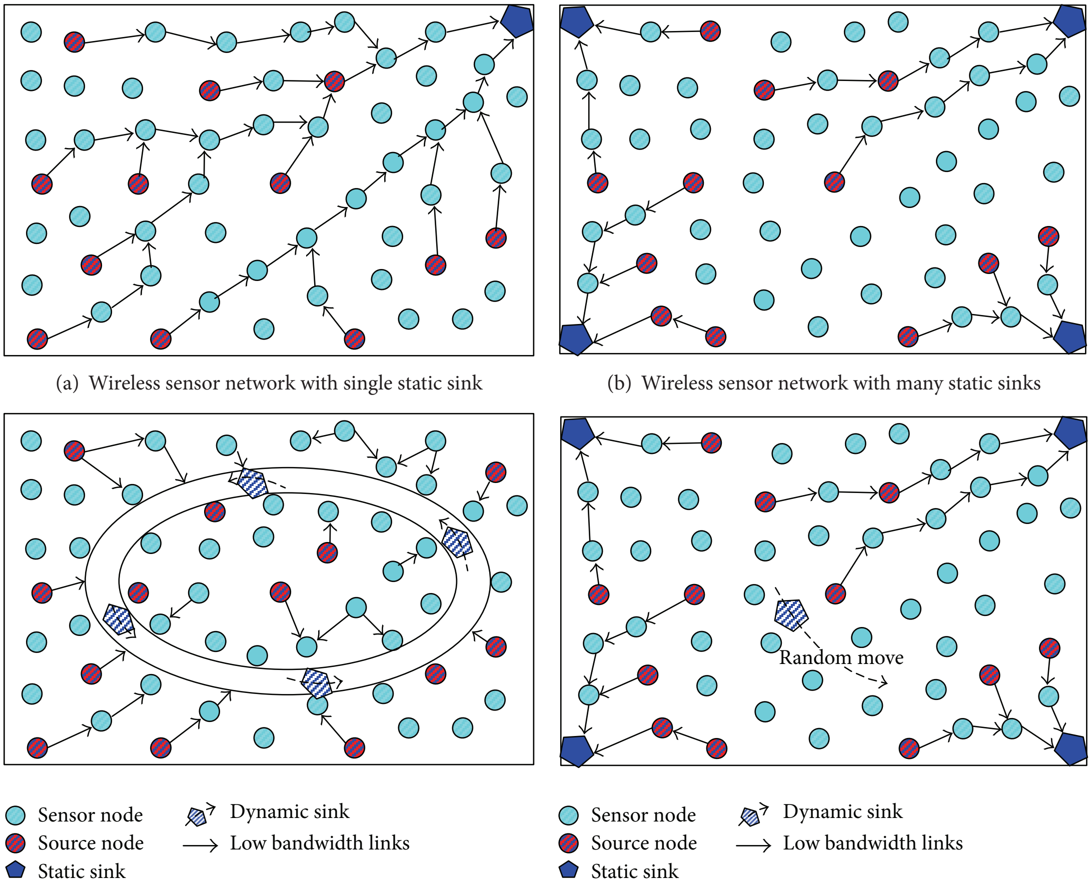
In addition, these network configurations increase the reliability and improve the network life time. However, leveraging the multisinks into the networks introduces new issues including the number of sinks, deploying mobile or static sinks, and hybrid sinks. Because the number of sinks has a great influence on the network performance, it should be studied carefully to find out the optimal number of sinks as a function of the network size. It is also required to study whether to (1) implement only static sinks as in Figure 6(b) which has four static sinks, as an example, distributed at the corners of the network field, (2) deploy only mobile nodes as depicted in Figure 6(c) which has four mobile sinks that move in a circular patterns as an example, or (3) leverage or combine both static and mobile sinks into the network field as shown in Figure 6(d) which has four static sinks and one mobile sink. As an example in this aspect of mobility, Suganthi et al. [70] deployed the multisinks technique in which they use three mobile sinks along with a static sink. They confine one mobile sink to circular pattern mobility, the second one to rectangular pattern mobility, and the other one to constant speed mobility. They measure the remaining energy of the deployed nodes without sink mobility and with sink mobility. The authors revealed that the remaining energy of the nodes with sink mobility is better than the remaining energy of nodes without sink mobility. This extends the life time of the deployed network. However, forwarding metrics for moving the mobile sinks and optimal number of sinks need more explorations and studies. Furthermore, the securely underlying scenarios should be considered. Thus, implementing multisinks into network fields can be considered as a new area of exploration.

6.4. Dynamic Holes Bypassing.

6.4. Dynamic Holes Bypassing. Dynamic holes occur due to overutilizing optimal paths. Figure 7(a) reveals that when a source node A always sends its data over an optimal path to the sink, it may result in energy holes as depicted in Figure 7(b). An energy hole or energy gap is a region in the network field that is not covered with radio signals. The increasing number of energy gaps reduces network life time and may render network ineffective. Therefore, the rest of this section presents possible bypassing solutions that elevate the problems resulted from energy holes.

Most of the current routing protocols such as [19, 20, 23–25, 28–30, 58] bypass energy holes by exploring new path to the sink node as shown in Figure 8(a), but it may result in an extra delay.

This solution works fine when a source node (e.g., node A) has a connection with the sink node but it fails when a source node is disconnected from the sink node. For example, the nodes surrounded by a dash,ed circle form a network partition, and they cannot deliver their data to the sink node.



(a) Wireless sensor network with single static sink

(b) Wireless sensor network with many static sinks

(c) Wireless sensor network with multidynamic sinks

(d) Wireless sensor network with hybrid sinks

Figure 6: Different multisinks and multisources techniques.

To overcome this shortcoming, the network dynamic should be implemented into networks. The network dynamic can be introduced into the network by implementing one of the following three aspects:

(1) small set of mobile nodes with a static sink,

(2) static nodes with mobile sink,

(3) small set of mobile nodes, mobile sink, and static sink.

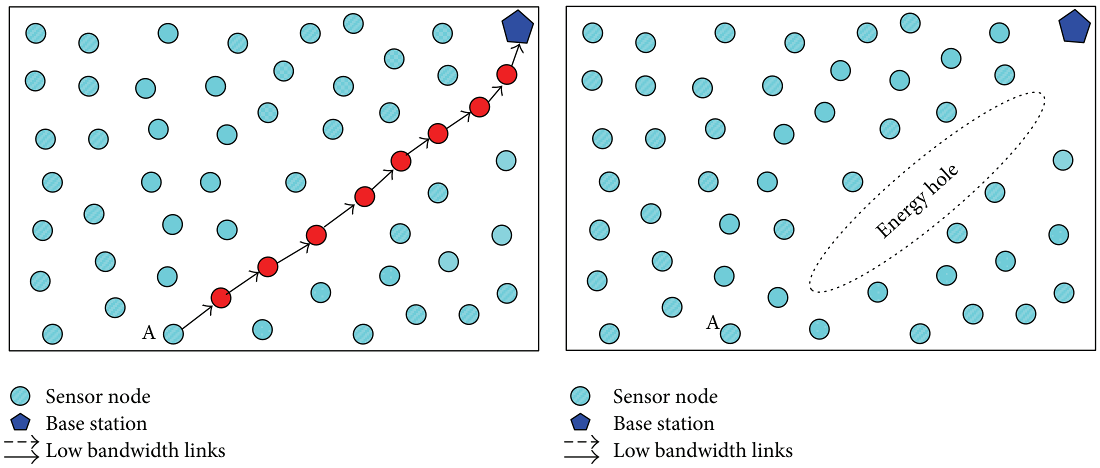
Implementing a small set of mobile nodes into a network field can cover energy hole regions with radio signals by moving one or more mobile nodes towards these regions as depicted in Figure 8(b). This solution introduces new issues that need a further consideration and study. These issues are summarized as follows: which mobile nodes should be moved towards energy hole regions, when they should be moved, and where exactly they should be moved. For example, which of the nodes B, C, or D in Figure 8(b) will be moved towards the indicated region? One solution to these issues can be achieved by giving a weight to each mobile node. The weight should be a function of the distance between the mobile node and the center of the region to be covered. Also, the weight should consider the priority of each energy gap region. In addition, it should have a concern about the current area covered by the mobile node.

The second aspect is to implement a mobile sink with static nodes. The mobile sink will move around to collect data from all sensors in the network field as seen in Figure 8(c) in which the sink goes into the region to collect the data. This will bypass the energy hole but introduces an extra overhead due to the updates concerning new location information. In addition, the mobile sink movement pattern should be defined as discussed in the network dynamics section.

The last aspect is to combine the static sink, mobile sink, mobile nodes, and static nodes into network field as shown in Figure 8(d) as an example. In this aspect, a source node has more choices to bypass an energy hole varying from exploring new route to adjust topology changes required to cover the holes. However, the performance- and security-related issues resulted from combining these different technologies need extra studies and exploration. In addition, more studies are needed to know how to adjust topology changes resulted from holes.

6.5. *Secure Routing.*

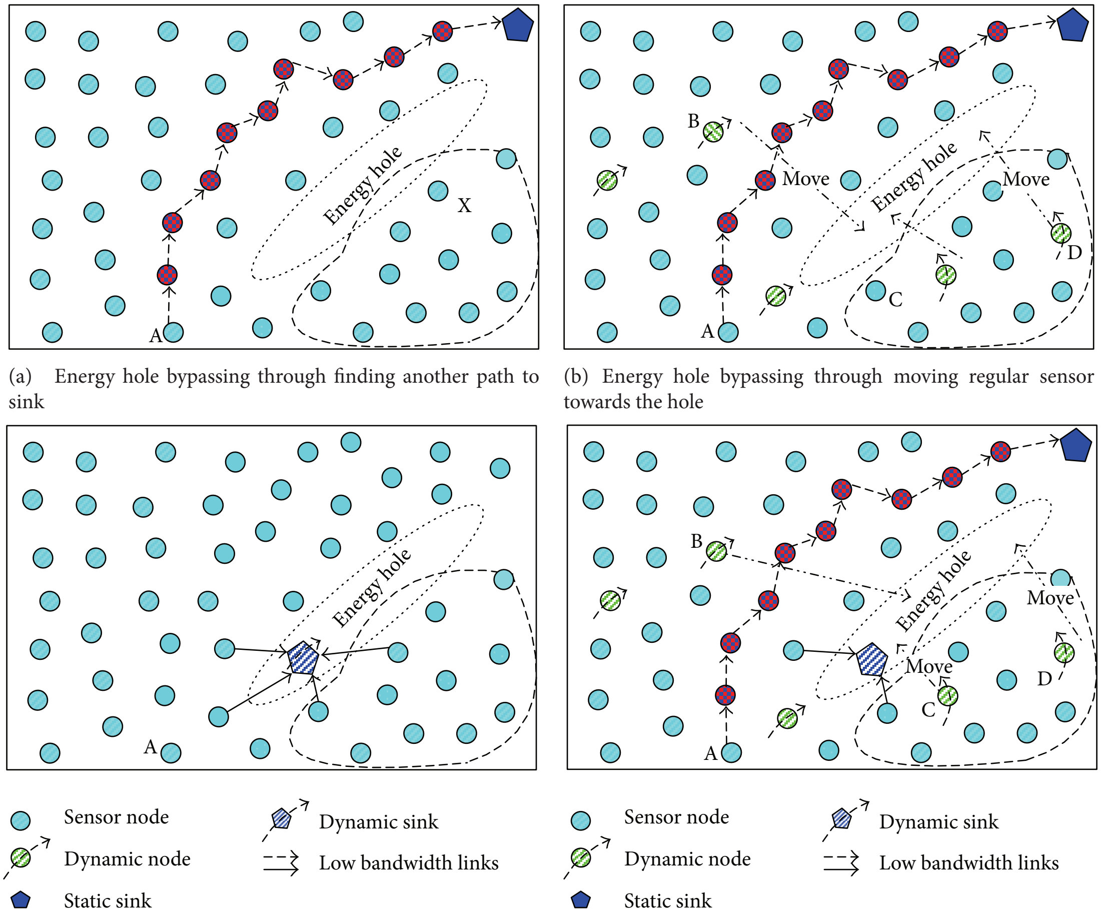
WSNs are used in several applications such as homeland security, military, and healthcare in which the security is essential. Such networks are subject to several routing attacks which include sinkhole attack, selective forwarding attacks, wormhole attack, Hello flood attack, Sybil



(a) Overutilization of energy efficiency

(b) Energy hole

FIGURE 7: Energy efficiency and energy holes.



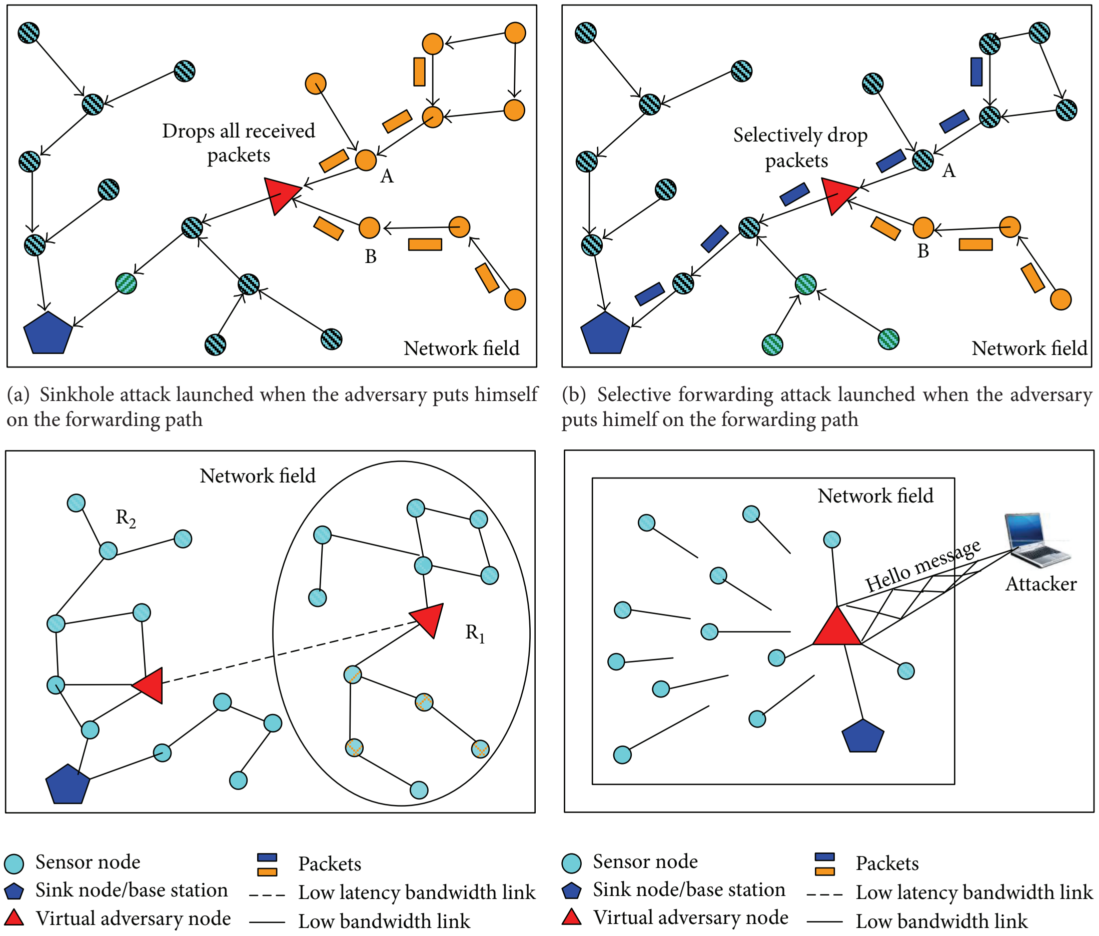
(a) Energy hole bypassing through finding another path to sink

(b) Energy hole bypassing through moving regular sensor towards the hole

(c) Energy hole bypassing through deploying a mobile sink

(d) Energy hole bypassing through hybrid mobility of both sink and regular sensors

FIGURE 8: Different techniques for energy holes bypassing.



(a) Sinkhole attack launched when the adversary puts himself on the forwarding path

(b) Selective forwarding attack launched when the adversary puts himelf on the forwarding path

(c) Wormhole attack using two adversaries

(d) Hello flood attack scenario using a laptop

Figure 9: Different routing attacks on WSNs.

attack, Denial-of-service (DoS) attack, and a sink replication attack associated with mobile sinks. Figure 9 depicts different types of routing attacks.

Figure 9(a) shows a sinkhole attack scenario that is launched when an adversary puts himself on the forwarding path. Once the adversary is on the forwarding path, he will refuse to forward any received packets. Figure 9(b) depicts the selective forwarding attack in which the attacker puts himself on the forwarding path and then drops certain packets and prevents them from reaching the sink node. Figure 9(c) visualizes the wormhole attack launched with two adversaries that share the low latency link denoted by a dashed line. One adversary attracts nodes in its neighborhood to send their data to the adversary. When the adversary receives data, he will forward the received data to the other adversary which in turn conveys it to the nodes in its region. Figure 9(d) depicts the Hello flood attack which is launched when an attacker, equipped with a laptop that has a high-quality radio signal, sends Hello messages to the network field. Once a node receives a Hello message, it marks the attacker as its parent. When the node has data that it wants to share, it sends it to the attacker, but because the attacker is far away, the node sends its data into oblivion [46, 71].

Another type of WSN attack on routing protocols is Sybil attack which reduces the fault tolerance, topology maintenance, and resource utilization and weakens the routing mechanism. In this type of attack, a node steals the identities of many nodes to pretend to be as them to degrade the data integrity and security of the network. But in reality, it is only a single malicious node injected cleverly into the specific sensor network to alter valuable information [47, 71].

The DoS attack in WSNs can occur intentionally by attackers or unintentionally due to unexpected node failure. The unexpected node failure may result from various software bugs, exhausted power supply system, environmental disaster, and complication in data transmission and communication, or even intentional intruder attack may execute DoS attack. Often, attackers try to weaken or destroy a network or cause an interruption in secure data communication by sending loads of unnecessary data packets to the victim nodes and therefore exhibit DoS attack. Different types of DoS attacks may take place at different network layers. At physical layer, it may cause jamming and tampering, at data link layer, it causes exhaustion and data collision, at network layer, it causes misdirection and negligence of data, and at transport layer, it could perform data flooding and malicious attack [71].

Finally, the sink replication attack is associated with sink mobility. It is achieved when a malicious sink attracts sensor nodes to forward their data to it. The aforementioned attacks may render the network inefficient. Therefore, several secure routing protocols such as [53–59, 72] have been proposed to defend against these attacks. However, new applications and new advance in the technology may need new security algorithms to protect networks against the current and new attacks that may be introduced. Thus, the security is an endless area of research, and integrating security into routing techniques of WSNs is an important area of current and future research.

7. Future Research Direction

Based on the recent trends discussed in Section 6, routing protocols that implement network dynamics initially contribute in solving energy holes bypassing problems and have a promising performance over routing protocols that implement static networks. Also, the routing protocols that use multisink outperform the routing protocols that deploy only a single sink. However, implementing these technologies into routing protocols introduces security holes. Therefore, the future research directions on WSNs routing protocols should focus on implementing multisink and nodes mobility along with satisfying security and QoS requirements into the new routing protocols. These routing protocols will help to protect their networks and prolong their life time.

Acknowledgment

This work is supported by the Deanship of Scientific Research, Taibah University, KSA under the Grant no. 434/4301.

References

[1] D. Culler, “Secure, low-power, IP-based connectivity with IEEE 802.15.4 wireless networks,” Industrial Embedded Systems, 2007.

[2] Z. Alliance, Zigbee alliance, http://www.zigbee.org.

[3] G. Mulligan, “The 6LoWPAN architecture,” in *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets ’07)*, pp. 78–82, http://portal.acm.org/citation.cfm?id=1278972.1278992.

[4] J. Zheng and A. Jamalipour, *Wireless Sensor Networks: A Networking Perspective*, Wiley/IEEE, 2009.

[5] T. J. Dishongh and M. McGrath, *Wireless Sensor Networks for Healthcare Applications*, Artech House, 2009.

[6] Y. Li, M. T. Thai, and W. Wu, *Wireless Sensor Networks and Applications*, Springer, 2008.

[7] H. Bai, M. Atiquzzaman, and D. Lilja, “Wireless sensor network for aircraft Health Monitoring,” in *Proceedings of the 1st International Conference on Broadband Networks (Broadnets ’04)*, pp. 748–750, San José, Calif, USA, October 2004.

[8] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, “Wireless sensor networks for habitat monitoring,” in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, Atlanta, Ga, USA, September 2002.

[9] G. Werner-Allen, K. Lorincz, M. Ruiz et al., “Deploying a wireless sensor network on an active volcano,” *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, 2006.

[10] R. Hongliang, M. Q. H. Meng, and X. Chen, “Physiological information acquisition through wireless biomedical sensor networks,” in *Proceedings of IEEE International Conference on Information Acquisition*, pp. 483–488, Macau, China, June 2005.

[11] S. Hengstler and H. Aghajan, “Application development in vision-enabled wireless sensor networks,” in *Proceedings of the 2nd International Conference on Systems and Networks Communications (ICSNC ’06)*, pp. 30–35, November 2006.

[12] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[13] S. K. Singh, M. P. Singh, and D. K. Singh, “A survey of energy-efficient hierarchical luster-based routing in wireless sensor networks,” *International Journal of Advanced Networking and Applications*, vol. 2, no. 2, pp. 570–580, 2010.

[14] K. Padmanabhan and P. Kamalakkannan, “A study on energy efficient routing protocols in wireless sensor networks,” *European Journal of Scientific Research*, vol. 60, no. 4, pp. 517–529, 2011.

[15] N. Pantazis, S. Nikolidakis, and D. Vergados, “Energy-efficient routing protocols in wireless sensor networks: a survey,” *IEEE of Communication Surveys and Tutorials*, vol. 15, no. 2, pp. 551–591, 2013.

[16] S. Ehsan and B. Hamdaoui, “A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks,” *IEEE Communications Surveys and Tutorials*, vol. 14, no. 2, pp. 265–278, 2012.

[17] A. M. Popescu, G. I. Tudorache, B. Peng, and A. H. Kemp, “Surveying position based routing protocols for wireless sensor and ad-hoc networks,” *International Journal of Communication Networks and Information Security*, vol. 4, no. 1, pp. 41–67, 2012.

[18] A. Triviño-Cabrera and S. Cañadas, “Survey on opportunistic routing in multihop wireless networks,” *International Journal of Communication Networks and Information Security*, vol. 3, no. 2, pp. 170–177, 2011.

[19] 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.

[20] W. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom ’99)*, pp. 174–185, Seattle, Wash, USA, August 1999.

[21] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks,” in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM ’00)*, pp. 56–67, August 2000.

[22] D. Braginsky and D. Estrin, “Rumor routing algorithm for sensor networks,” in *Proceedings of the first Workshop on Sensor Networks and Applications (WSNA ’02)*, Atlanta, Ga, USA, October 2002.

[23] W. R. Heizelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless micro sensor networks,” in *Proceedings of IEEE Hawaii International Conference on System Sciences*, pp. 3005–3024, Washington, DC, USA, January 2000.

[24] A. Manjeshwar and D. P. Agarwal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks," in *Proceedings 15th International in Parallel and Distributed Processing Symposium*, pp. 2009–2015, April 2001.

[25] S. Lindsey and C. S. Raghayendra, "PEGASIS: power-efficient gathering in sensor information system," in *Proceedings of IEEE Aerospace Conference*, vol. 3, pp. 1125–1130, 2002.

[26] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, 2000.

[27] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: a recursive data dissemination protocol for WSNs," Technical Report UCLA-CSD TR-01-0023, University of California at Los Angeles Computer Science Department, May 2001.

[28] S. S. Chiang, C. H. Huang, and K. C. Chang, "A minimum hop routing protocol for home security systems using wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 4, pp. 1483–1489, 2007.

[29] M. M. A. Azim, "MAP: energy efficient routing protocol for wireless sensor networks," *Journal of Information Processing Systems*, vol. 6, no. 3, pp. 295–316, 2010.

[30] A. M. El-Semary and M. M. A. Azim, "Path energy weight: a global energy-aware routing protocol for Wireless sensor Networks," in *Proceedings of IEEE Conference on IFIP Wireless Days (WD '10)*, Venice, Italy, October 2010.

[31] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960, 2007.

[32] S. Ehsan and B. Hamdaoui, "A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 2, pp. 265–278, 2012.

[33] T. Shu, M. Krunz, and S. Liu, "Secure data collection in wireless sensor networks using randomized dispersive routes," *IEEE Transactions on Mobile Computing*, vol. 9, no. 7, pp. 941–954, 2010.

[34] A. Proano and L. Lazos, "Packet-hiding methods for preventing selective jamming attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 101–114, 2012.

[35] G. S. Murthy, R. J. D'Souza, and G. Varaprasad, "Digital signature-based secure node disjoint multipath routing protocol for wireless sensor networks," *IEEE Sensors Journal*, vol. 12, no. 10, pp. 2941–2949, 2012.

[36] A. Liu, Z. Zheng, C. Zhang, Z. Chen, and X. Shen, "Secure and energy-efficient disjoint multipath routing for WSNs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 7, pp. 3255–3265, 2012.

[37] A. Modirkhazeni, N. Ithnin, and O. Ibrahim, "Secure multipath routing protocols in wireless sensor networks: a security survey analysis," in *Proceedings of the 2nd International Conference on Network Applications, Protocols and Services (NETAPPS '10)*, pp. 228–233, September 2010.

[38] T. Halim and M. R. Islam, "A study on the security issues in WSN," *International Journal of Computer Application*, vol. 53, no. 1, pp. 0975–8887, 2012.

[39] S. Khan, N. Mast, K. K. Loo, and A. Silahuddin, "Passive security threats and consequences in IEEE 802.11 wireless mesh networks," *International Journal of Digital Content Technology and Its Application*, vol. 2, no. 3, pp. 4–8, 2008.

[40] S. Khan, N. A. Alrajeh, and K. K. Loo, "Secure route selection in wireless mesh networks," *Computer Networks*, vol. 56, no. 2, pp. 491–503, 2012.

[41] T. Zia and A. Y. Zomaya, "Security issues and countermeasures in wireless sensor networks," in *Algorithms and Protocols for Wireless Sensor Networks*, John Wiley & Sons, 2009.

[42] M. Healy, T. Newe, and E. Lewis, "Security for wireless sensor networks: a review," in *Proceedings of IEEE Sensors Applications Symposium (SAS '09)*, pp. 80–85, New Orleans, La, USA, February 2009.

[43] C. Gupta, K. Gupta, and V. Gupta, "Security threats in sensor network and their possible solutions," in *Proceedings of the International Symposium on Instrumentation and Measurement, Sensor Networks, and Automation (IMSNA '12)*, pp. 25–28, Sanya, China, August 2012.

[44] Y. C. Hu and A. Perrig, "Wormhole attacks in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 370–379, 2006.

[45] W. Z. Khan, Y. Xiang, and M. Y. Aalsalem, "Comprehensive study of selective forwarding attack in wireless sensor networks," *International Journal on Computer Network and Information Security*, vol. 1, pp. 1–10, 2011.

[46] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[47] J. R. Douceur, "The sybil attack," in *Proceedings of the1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, March 2002.

[48] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 2, pp. 228–258, 2005.

[49] Q. Dong and D. Liu, "Using auxiliary sensors for pairwise key establishment in wsn," in *Networking*, I. F. Akyildiz, R. Sivakumar, Eylem Ekici, J. C. de Oliveira, and J. McNair, Eds., vol. 4479 of *Lecture Notes in Computer Science*, pp. 251–262, Springer, 2007.

[50] R. Sharma, Y. Chaba, and Y. Singh, "An IPC key management scheme for wireless sensor network," in *Proceedings of the 1st IEEE International Conference on Parallel, Distributed and Grid Computing (PDGC '10)*, pp. 251–255, October 2010.

[51] I. S. Gawdan, C. Chow, A. Tanveer Zia, and Q. I. Sarhan, "A novel secure key management module for hierarchical clustering wireless sensor networks," in *Proceedings of IEEE 3rd International Conference on Computational Intelligence, Modelling and Simulation*, pp. 312–316, 2011.

[52] S. Ruj, A. Nayak, and I. Stojmenovic, "Pairwise and triple key distribution in wireless sensor networks with applications," *IEEE Transactions on Computers*, no. 99, p. 1.

[53] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: security protocols for sensor networks," in *Proceedings of 7th annual International Conference on Mobile Computing and Networking (MobiCom'01)*, pp. 189–199, Rome, Italy, July 2001.

[54] X. Debao, W. Meijuan, and Z. Ying, "Secure-SPIN: secure sensor protocol for information via negotiation for wireless sensor networks," in *Proceedings of the 1st IEEE Conference on Industrial Electronics and Applications (ICIEA '06)*, May 2006.

[55] J. Zhou, C. Li, Q. Cao, and Y. Shen, "An Intrusion-tolerant secure routing protocol with key exchange for wireless sensor network," in *Proceedings of IEEE International Conference on*

Information and Automation (ICIA '08), pp. 1547–1552, June 2008.

[56] Y. Zhang, J. Yang, W. Li, L. Wang, and L. Jin, "An authentication scheme for locating compromised sensor nodes in WSNs," *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 50–62, 2010.

[57] A. M. El-Semary, "Energy-efficient secure routing protocol based on rulette-wheel and μTESLA for wireless sensor networks," *The International Journal of Sensor Networks and Data Communications*, vol. 1, Article ID X110201, 13 pages, 2012.

[58] M. Aly El-Semary and A. Mohamed Azim, "A dual-sink secure routing protocol for wireless sensor networks," *Journal of Information Assurance and Security*, vol. 7, pp. 324–337, 2012.

[59] A. Rasheed and R. N. Mahapatra, "The three-tier security scheme in wireless sensor networks with mobile sinks," *IEEE Transaction on Parallel and Distributed Systems*, vol. 23, no. 5, pp. 958–965, 2012.

[60] Y. X. Jiang and B. H. Zhao, "A secure routing protocol with malicious nodes detecting and diagnosing mechanism for wireless sensor networks," in *Proceedings of IEEE Asia-Pacific Services Computing Conference (APSCC '07)*, pp. 49–55, IEEE Computer Society, December 2007.

[61] N. Gui, J. Hu, and Z. Chen, "A secure routing protocol and its application in multi-sink wireless sensor networks," *Journal of Networks*, vol. 5, no. 5, pp. 535–542, 2010.

[62] P. Ning and D. Liu, "Broadcast authentication and key management for secure sensor networks," in *Handbook of Sensor Networks: Algorithms and Architectures*, I. Stojmenović, Ed., John Wiley & Sons, 2005.

[63] V. T. Kesavan and S. Radhakrishnan, "Multiple secret keys based security for wireless sensor networks," *International Journal of Communication Networks and Information Security*, vol. 4, no. 1, pp. 68–76, 2012.

[64] S. Khan, K. K. Loo, N. Mast, and T. Naeem, "SRPM: secure routing protocol for IEEE 802.11 infrastructure based wireless mesh networks," *Journal of Network and Systems Management*, vol. 18, no. 2, pp. 190–209, 2010.

[65] S. Khan, K. K. Loo, and Z. U. Din, "Cross layer design for routing and security in multi-hop wireless networks," *Journal of Information Assurance and Security*, vol. 4, pp. 170–173, 2009.

[66] S. Khan and J. Loo, "Cross layer secure and resource-aware on-demand routing protocol for hybrid wireless mesh networks," *Wireless Personal Communications*, vol. 62, no. 1, pp. 201–214, 2010.

[67] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and Z. M. Wang, "Controlled sink mobility for prolonging wireless sensor networks lifetime," *Wireless Networks*, vol. 14, no. 6, pp. 831–858, 2008.

[68] X. Liu, H. Zhao, X. Yang, and X. Li, "SinkTrail: a proactive data reporting protocol for wireless sensor networks," *IEEE Transactions on Computers*, vol. 62, no. 1, pp. 151–162, 2013.

[69] M. Ma and Y. Yang, "Data gathering in wireless sensor networks with mobile collectors," in *Proceedings of the 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS '08)*, pp. 1–9, April 2008.

[70] K. Suganthi, B. V. Sundaram, K. S. Kumar, J. S. Ashim, and S. S. Kumar, "Improving energy efficiency and reliability using multiple mobile sinks and hierarchical clustering in wireless sensor networks," in *Proceedings of IEEE-International Conference on Recent Trends in Information Technology (ICRTIT '11)*, MIT, Anna University, Chennai, India, June 2011.

[71] T. Halim and R. Islam, "A study on the security issues in WSN," *International Journal of Computer Applications*, vol. 53, no. 1, p. 26, 2012.

[72] S. P. Srinivasan and C. Chellappan, "Semi-randomised propagation for secure routing in wireless sensor networks," in *Proceedings of IEEE-International Conference on Recent Trends in Information Technology (ICRTIT '11)*, MIT, Anna University, Chennai, India, June 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 167575, 7 pages
http://dx.doi.org/10.1155/2013/167575

Research Article

Intrusion Detection Systems in Wireless Sensor Networks: A Review

Nabil Ali Alrajeh,[1] S. Khan,[2] and Bilal Shams[2]

[1] *Biomedical Technology Department, College of Applied Medical Sciences, King Saud University, Riyadh 11633, Saudi Arabia*
[2] *Institute of Information Technology, Kohat University of Science and Technology (KUST), Kohat City 26000, Pakistan*

Correspondence should be addressed to Nabil Ali Alrajeh; nabil@ksu.edu.sa

Received 28 February 2013; Accepted 16 April 2013

Academic Editor: Jaime Lloret

Wireless Sensor Networks (WSNs) consist of sensor nodes deployed in a manner to collect information about surrounding environment. Their distributed nature, multihop data forwarding, and open wireless medium are the factors that make WSNs highly vulnerable to security attacks at various levels. Intrusion Detection Systems (IDSs) can play an important role in detecting and preventing security attacks. This paper presents current Intrusion Detection Systems and some open research problems related to WSN security.

1. Introduction

Wireless Sensor Networks (WSNs) are composed of sensor nodes and sinks. Sensor nodes have the capability of self-healing and self-organizing. They are decentralized and distributed in nature where communication takes place via multihop intermediate nodes. The main objective of a sensor node is to collect information from its surrounding environment and transmit it to the sink. WSNs have many applications and are used in scenarios such as detecting climate changed, monitoring environments and habitats, and various other surveillance and military applications. Mostly sensor nodes are used in such areas where wired networks are impossible to be deployed. WSNs are deployed in physical harsh and hostile environments where nodes are always exposed to physical security risks damages. Furthermore, self-organizing nature, low battery power supply, limited bandwidth support, distributed operations using open wireless medium, multihop traffic forwarding, and dependency on other nodes are such characteristics of sensor networks that expose it to many security attacks at all layers of the OSI model.

Many security-related solutions for WSNs have been proposed such as authentication, key exchange, and secure routing or security mechanisms for specific attacks. These security mechanisms are capable of ensuring security at some level; however they cannot eliminate most of the security attacks [1]. An IDS is one possible solution to address a wide range of security attacks in WSNs.

An IDS is also referred to as a second line of defence, which is used for intrusion detection only; that is, IDS can detect attacks but cannot prevent or respond. Once the attack is detected, the IDSs raise an alarm to inform the controller to take action. There are two important classes of IDSs. One is rule-based IDS and the other is anomaly-based IDS [2, 3]. Rule-based IDS is also known as signature-based IDS which is used to detect intrusions with the help of built-in signatures. Rule-based IDS can detect well-known attacks with great accuracy, but it is unable to detect new attacks for which the signatures are not present in intrusion database. Anomaly-based IDSs detect intrusion by matching traffic patterns or resource utilizations. Although anomaly based IDSs have the ability to detect both well-known and new attacks, they have more false positive and false negative alarms. Some IDSs operate in specific scenarios or with particular routing protocols. Watchers [4] operate with proactive routing protocol to detect routing anomalies. It is implemented on each node, so all the nodes need some sort of cooperation to detect routing intrusions. Some intrusion detection mechanisms also operate with reactive routing protocols [5, 6]. These mechanisms enable the network to select a reliable path from source to destination.

This paper presents a review of existing IDSs. It is organized as follows. In Section 2, we examine existing security attacks. In Section 3, we analyze and discuss some already proposed IDSs. We make comparison of existing IDSs on the basis of detection. In Section 4, we highlight some open research issues and directions, and finally in Section 5, we conclude the paper.

2. Overview of Security in Wireless Sensor Networks

WSNs are vulnerable to many types of security attacks due to open wireless medium, multihop decentralized communication, and deployment in hostile and physically nonprotected areas. Different threat models are discussed in [7] such as mote-class attacks and laptop-class attacks. In mote-class attacks, the attacker compromises few of the sensor nodes inside a WSN. In laptop-class attacks, the attacker has more powerful device(s) to launch more intense attack against WSNs.

Security attacks against WSNs can be classified as active and passive [8–10]. Passive attacks are silent in nature and are conducted to extract important information from the network. Passive attacks do not harm the network or network resources. Active attacks are used to misdirect, temper, or drop packets. The unique characteristics such as wireless medium, contention-based medium access, multihop nature, decentralized architecture, and random deployment of such networks make them more vulnerable to security attacks at various layers.

Physical layer of WSN is responsible for radio and signals management. Radio jamming is one of the severe attacks against WSN [8, 11]. Another physical layer attack is battery exhaustion attack. In a WSN, battery power of sensor nodes plays an important role and determines the lifetime of the network. Keeping in view the power limitations of WSNs, it is highly desirable to design power efficient mechanisms for sustainable WSNs. Sensor nodes in sleep mode consume less energy as compared to active mode. In energy exhaustion attack, the attacker tries not to allow sensor nodes to switch to sleep mode. This can be done by sending unnecessary data or beacons to sensor nodes to keep them always busy. As WSNs are deployed in hostile environment, it is susceptible to many physical attacks such as node destruction, node replacement, node replication, battery replacement, or reprogramming of node with malicious code [12, 13]. However such attacks need to physically access the network.

Most WSNs use contention based carrier sense multiple access with collision avoidance mechanism (CSMA/CA). This mechanism tries to avoid collision; however it adds more complications in the form of collision, hidden-node problem, MAC selfishness, and unfairness [7, 8]. Possible countermeasures against such kind of attacks are small frames and rate limitations [7, 14].

Network layer is responsible for appropriate route selection from source to destination [15, 16]. In WSN, the multihop route from source to destination is vulnerable to many active and passive attacks [17, 18]. Active attacks include packet-dropping attacks, packet-misdirecting attacks, rushing attack, Sybil attack, byzantine attack, routing table overflow attack, spoofed routing information, hello flood, and acknowledgement spoofing [8, 19].

3. Intrusion Detection Systems

One of the key features of a WSN is its multihop distributed operations, which add more complexity in terms of security attack detection and prevention. In a multihop distributed environment, it is very difficult to locate attackers or malicious nodes. Many security attack detection and prevention mechanisms are designed for WSNs; however most of the existing solutions are capable of handling only a few security attacks. For example, most secure routing protocols are designed to counter few security attacks [20, 21]. Similarly new media access mechanisms are designed to handle hidden-node problem or selfishness. Encryption mechanisms are designed to protect data against passive attacks. Hence, one can say that there is a need to design mechanisms that are capable enough of detecting and preventing multiple security attacks in WSNs. An Intrusion Detection System (IDS) is one possible solution to it.

An intrusion is basically any sort of unlawful activity which is carried out by attackers to harm network resources or sensor nodes. An IDS is a mechanism to detect such unlawful or malicious activities [22]. The primary functions of IDS are to monitor users' activities and network behaviour at different layers.

A single perfect defence is neither feasible nor possible in wireless networks, as there always exist some architectural weaknesses, software bugs, or design flaws which may be compromised by intruders. The best practice to secure wireless networks is to implement multilines of security mechanisms; that is why IDS is more critical in wireless networks. It is viewed as a passive defence, as it is not intended to prevent attacks; instead it alerts network administrators about possible attacks well in time to stop or reduce the impact of the attack. The accuracy of intrusion detection is generally measured in terms of false positives (false alarms) and false negatives (attacks not detected), where the IDSs attempt to minimize both these terms [3].

There are two important classes of IDSs. One is known as signature-based IDS, where the signatures of different security attacks are maintained in a database. This kind of IDS is effective against well-known security attacks. However, new attacks are difficult to be detected as their signatures would not be present in the database. The second type is anomaly-based IDS. This kind is effective to detect new attacks; however it sometimes misses to detect well-known security attacks. The reason is that anomaly-based IDSs do not maintain any database, but they continuously monitor traffic patterns or system activities.

IDS can operate in many modes, for example, standalone operation and cooperative cluster based operation [23]. A standalone IDS operates on every node to detect unwanted activities. Cooperative cluster based IDS are mostly distributed in nature in which every node monitors its neighbours and surrounding nodes activities and operation; in case of any malicious activity detection, the cluster head is informed.

Broadly speaking, IDS has three main components [3] as shown in Figure 1.

(i) Monitoring component is used for local events monitoring as well as neighbours monitoring. This component mostly monitors traffic patterns, internal events, and resource utilization [24].

(ii) Analysis and detection module is the main component which is based on modeling algorithm. Network operations, behavior, and activities are analyzed, and decisions are made to declare them as malicious or not.

(iii) Alarm component is a response generating component, which generates an alarm in case of detection of an intrusion.

It should be noted that IDSs are passive in nature and can only detect intrusion. They cannot take any preventive action; they only generate an alarm. It is then the administrator's job to take preventive measures against the attack. Researchers in WSNs are working on two broad categories of IDSs, that is, signature-based and anomaly-based IDSs.

3.1. Signature-Based Intrusion Detection Systems.

Signature-based IDS, also known as rule-based IDS, has predefined rules of different security attacks. When the network's behaviour shows any deviation from the predefined rules, it is classified as an attack. Signature-based IDSs are well suited for known intrusions; however they cannot detect new security attacks or those attacks having no predefined rules [3]. In this section, we present existing signature-based IDSs for WSNs.

In [25], a rule-based IDS for WSNs is presented. It is host based in which every node has IDS. The architecture of the proposed IDS has many modules such as packet monitoring, cooperative engine, detection engine, and response unit. The IDS is basically designed for routing attacks and is capable of detecting packet-dropping attacks. An IDS for detection of sink-hole attack is presented in [26]. The proposed IDS is hosted on each sensor node and requires TinyOS with the combination of MintRoute routing protocol. It is an advanced version of [25] with narrow approach; that is, the former can detect many packet-dropping and misdirecting attacks while the latter is only designed for detection of sink-hole attacks. In both approaches, every node monitors and cooperates with neighbours. Intrusion Detection Architecture (IDA) is presented in [27]. IDA is distributed and hierarchical in nature which can operate by cooperation of sensor nodes, cluster head, and central system. IDA generates either passive or active response on the basis of attack nature. However, this work does not present results on the detection rate and false positive and false negative ratios.

In [28], Intrusion Detection Program (IDP) is proposed, which is capable to detect known attacks. IDP is based on genetic programming (GP) technique and is effective against a variety of attacks such as denial of service (DoS) and unauthorized access. IDA uses three variants of GP such as linear-genetic programming (LGP), multiexpression programming (MEP), and gene-expression programming (GEP). GEP and MEP detection and classification accuracy are greater than 95%. A distributed IDS (DIDS) using soft computing techniques is presented in [29]. It uses few fuzzy rule-based classifiers to identify intrusions. The authors claim that fuzzy classifier provides 100% accuracy for all kinds of intrusions.

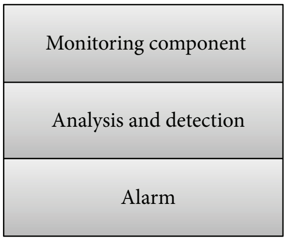


Figure 1: Components of IDS.

A decentralized rule-based IDS is proposed in [30]. This mechanism has three main phases, namely, data acquisition, rule application, and intrusion detection. The proposed mechanism is capable of detecting many routing attacks such as worm-hole, black-hole, selective-forwarding, and delay attacks. The authors also claim that the proposed solution is capable of detecting jamming attack as well; however they did not explain how jamming attacks are detected as it is a physical layer attack. Spontaneous watchdog IDS and its basic architecture is given in [31]. This architecture consists of local and global agents; however it is not implemented yet. An ant-colony-based IDS in conjunction with machine learning [32] is another rule-based IDS. The proposed IDS perceives behaviour and acts using self-organizing principle initiated with probability values. Different signature-based IDSs are given in Table 1.

3.2. Anomaly-Based Intrusion Detection Systems.

Anomaly-based IDS monitors network activities and classifies them as either normal or malicious using heuristic approach. Most of anomaly-based IDSs identify intrusions using threshold values; that is, any activity below a threshold is normal, while any condition above a threshold is classified as an intrusion. The main advantage of anomaly-based IDS is its capability to detect new and unknown attacks; however sometimes it fails to detect even well-known security attacks. Many anomaly-based IDSs have been proposed so far [33]. An unsupervised neural network based IDS [34] is capable of learning and detecting unknown attacks. This intelligent system learns the time-related changes using Markov model. When any intrusion occurs, a mobile agent moves to the malicious region of the WSN to investigate. The proposed mechanism can detect time-related changes and events.

A set of intrusion detection techniques at different layers is presented [35]. These techniques are independent of each other. At physical layer, RSSI values are used to detect masquerade, while at network layer, a specialized table driven routing protocol is used to detect routing and authentication

Table 1: Signature based IDSs.

IDS	Mechanism	Attacks	Evaluation metrics
[25]	Collaborative	Black hole, selective forwarding	Window length, false negative rates
[26]	Local and cooperative detection	Sink hole	Detection rate, false negative rates
[27]	Hierarchical	N/A	N/A
[28]	Genetic programming	DoS, unauthorized access	Classification accuracy
[29]	Soft computing	Unauthorized access, probing	Classification accuracy
[30]	Specification based	Repetition attack, delay attack, worm hole, alteration attack, black hole, selective forwarding	Detection rate, false positives
[31]	Spontaneous watchdog	N/A	N/A
[32]	Ant colony	Abnormal transmission	N/A

Table 2: Anomaly based IDSs.

IDS	Mechanism	Attacks
[34]	Artificial neural network	Time related changes
[35]	Set of techniques at OSI layers	Masquerade, routing attacks
[36]	Cluster based	Periodic route error attack, sink hole attack
[37]	Support vector	Black-hole attacks
[38]	Cross feature	Packet dropping attacks
[39]	Sliding window	Route depletion attack

Table 3: Hybrid IDSs.

IDS	Mechanism	Attacks
[40]	Hybrid, hierarchical	N/A
[41]	Support vector machine	N/A
[42]	State transition	Sync flood
[43]	Cluster based	Routing attacks
[44]	Cluster based, supervised learning, misuse detection	Routing attacks
[45]	Hierarchical and hybrid	Sink hole, worm hole

attacks. A cluster based IDS for routing attack is proposed [36]. This mechanism is capable of building a normal traffic model, which is used to differentiate between normal and abnormal traffic. The normal traffic model consists of number of packets received and sent, number of route requests received and sent, and so forth. The IDS can detect many attacks such as periodic route error attack and sink-hole attack. A support vector machine based IDS [37] is used to detect routing attacks such as black hole. It is basically cooperation based detection in which nodes communicate and share information about security attacks. A cross feature based anomaly detection mechanism is proposed in [38]. This mechanism monitors and learns normal traffic patterns in order to detect any intrusion in case of deviation. The IDS is capable of detecting packet-dropping and misdirecting attacks. A sliding window based IDS using threshold value is efficient in the detection of few security attacks such as route depletion attacks [39]. Table 2 presents a summary of a number of anomaly-based IDSs.

3.3. Hybrid Intrusion Detection Systems. Hybrid IDSs are a combination of both anomaly-based and signature-based approaches. Hybrid mechanisms usually contain two detection modules; that is, one module is responsible of detecting well-known attacks using signatures, while the other is responsible for detecting and learning normal and malicious patterns or monitor network behavior deviation from normal profile. Hybrid IDSs are more accurate in terms of attack detection with less number of false positives. However, such mechanisms consume more energy and more resources. Hybrid IDSs are generally not recommended for a resource constraint networks such as a WSN; however they are still an active research area. A hybrid intrusion detection model is presented in [40]. In this model, sensor nodes are divided into hexagonal regions like cellular networks. Each region is monitored by a cluster node, while cluster nodes are monitored by regional nodes. The base station has the responsibility to monitor all regional nodes. It is hierarchical in nature forming a tree-like structure. Attack signatures are stored in base station and propagated toward the leaf node for attack detection. Similarly the mechanism has predefined specifications of normal and abnormal behaviour. Anomaly detection is done by measuring deviation from defined specifications. The authors did not mention detection rate or false-alarm ratio of their proposed mechanism. Furthermore, it is not clear which security attacks are detected using this mechanism.

Another hybrid IDS using support vector machine (SVM) and misuse detection is proposed in [41]. A distributed learning algorithm is used to train SVM to distinguish normal and malicious patterns. This intrusion detection mechanism is designed to operate in cluster based WSNs, where all nodes monitor their neighbours. The authors claim high detection rate with fewer false positives; however attack types are not described. An IDS that uses state transition analysis and stream flow to detect sync-flood attack against WSNs is presented in [42]. This mechanism monitors three-way handshake of TCP to identify attack pattern; however it is not yet implemented and tested. A cluster based hybrid

Table 4: Comparison of different IDSs.

Characteristics	Anomaly based IDS	Signature based IDS	Hybrid IDS	Cross layer IDS
Detection rate	Medium	Medium	High	High
False alarm	Medium	Medium	Low	Low
Computation	Low	Low	Medium	High
Energy consumption	Low	Low	Medium	High
Attack detection	Few	Few	More	More
Multilayer attack detections	No	No	No	Yes
Strength	Capable of detecting new attacks	Detects all those attacks having signatures	Can detect both existing and new attacks	Can detect multilayer attacks
Weakness	Misses well known attack	Cannot detect new attacks	Requires more computation and resources	Requires more resources
Suitable for WSN	Yes	Yes	With justification	With strong justification

IDS is given in [43], where the cluster head is responsible for detecting intrusions. The key idea behind this mechanism is to reduce energy consumption. A further enhanced IDS is proposed in [44]. The enhanced IDS has three modules, that is, anomaly-based detection, signature-based detection, and decision making. A supervised back propagation network is used to learn and identify normal and malicious packets. Another hierarchical hybrid IDS for detection of routing attacks is presented in [45]. It has high accuracy in terms of detection of network layer security attacks such as sink hole and worm hole. Table 3 presents a summary of a few hybrid IDSs.

3.4. Cross Layer Intrusion Detection Systems. Cross layer design is a relatively new security technique in which different parameters across OSI layers are exchanged for optimal solutions [46]. Traditional IDS operates at a single layer of the OSI model and hence can monitor and detect intrusions at that particular layer. For example, network layer Intrusion Detection System can detect only routing attacks but cannot respond to MAC, physical, or transport layer anomalies. Cross layer IDSs have the capability to monitor and detect intrusions at multiple layers by communicating and exchanging parameters amongst different layers using cross layer interface. As we know, WSNs have many constraints in terms of computations, memory, and energy. Although cross layer IDS can detect many intrusions at different layers, this technique consumes more energy and computational resources by monitoring, analyzing, and exchanging multilayer parameters.

Cross layer intrusion detection agent (CLIDA) for WSNs is proposed in [47]. CLIDA ensures cross layer information exchange amongst physical, MAC, and network layer. Cross layer data module collects and represents data to all layers. CLIDA is capable of detecting multi-layer security attacks. This architecture has good detection rate; however energy and computational comparison is not given, which could be more interesting. Another cross layer security mechanism for WSN is proposed in [48], in which the authors have the observations that such mechanism would exhaust the limited resources of sensor nodes. In [24], a real-time cross layer security mechanisms for large scale flood detection and attack trace-back mechanism is presented. It uses different parameters from MAC and network layers to detect multilayer flooding attacks. It maintains different profiles for low, medium, and high intensity attacks.

4. Comparison and Discussion

Wireless Sensor Networks are distributed in nature using the multihop communication model. These networks are usually deployed in such areas where direct human interaction is either impossible or very difficult. Furthermore, WSNs have limitations in terms of computation, bandwidth, memory, and energy. These limitations are considered while designing any proposal for such networks. Due to the hostile environments of WSNs, security is one of their most important aspects. IDSs are widely used for securing WSNs. IDS has the ability to detect an intrusion and raise an alarm for appropriate action. Due to the energy and computational power limitations, designing appropriate IDS for WSN is a challenging task.

Anomaly-based IDSs are suitable for small-sized WSNs where few nodes communicate with the base station. In small sized WSNs, the traffic pattern is mostly the same, so unusual traffic pattern or changing behaviour can be treated as an intrusion. However such IDS may generate more false alarms and may not be able to detect well-known intrusions. Anomaly-based IDSs are usually lightweight in nature and mostly use statistical, probabilistic, traffic analysis or intelligent techniques.

Signature-based IDSs are suitable for relatively large-sized WSNs, where more security threats and attacks can compromise network operations. Signature-based IDS needs more resources and computations as compared to anomaly-based IDS. One of the important and complex activities is the compilation and insertion of new attack signatures in the databases. Such IDSs mostly use data mining or pattern matching techniques.

Hybrid IDSs are suitable for large and sustainable WSNs. These IDSs have both anomaly-based and signature-based modules, so they require more resources and computations.

To reduce the usage of limited resources, such mechanisms are mostly used in cluster based or hierarchical WSNs, in which some parts of the network are used to execute anomaly detection while other parts are accompanied with signature-based detection.

Cross layer IDSs are usually not recommended for a resource constraint networks such as WSNs, as it consumes more resources by exchanging parameters across the protocol suits for attack detection. Table 4 gives the comparison and characteristics of different IDSs.

5. Conclusions

While designing a security mechanism, we must consider the limited resources of WSNs. Anomaly-based IDSs are lightweight in nature; however they create more false alarms. Signature-based IDSs are suitable for relatively large-sized WSNs; however they have some overheads such as updating and inserting new signatures. Cross layer IDSs are usually not recommended for networks having resources limitations, as more energy and computation are required for exchanging multilayer parameters.

Acknowledgments

The authors extend their appreciation to the Research Centre, College of Applied Medical Sciences, and the Deanship of Scientific Research at King Saud University for funding this research.

References

[1] Y. Ping, J. Xinghao, W. Yue, and L. Ning, "Distributed intrusion detection for mobile ad hoc networks," *Journal of Systems Engineering and Electronics*, vol. 19, no. 4, pp. 851–859, 2008.

[2] S. Northcutt and J. Novak, *Network Intrusion Detection*, SAMS, 3rd edition, 2002.

[3] S. Khan, K. K. Loo, and Z. U. Din, "Framework for intrusion detection in IEEE 802.11 wireless mesh networks," *International Arab Journal of Information Technology*, vol. 7, no. 4, pp. 435–440, 2010.

[4] T. M. Chen, G.-S. Kuo, Z.-P. Li, and G.-M. Zhu, "Intrusion detection in wireless mesh networks," in *Security in Wireless Mesh Networks*, Y. Zhang, J. Zheng, and H. Hu, Eds., CRC Press, New York, NY, USA, 2007.

[5] M. K. Rafsanjani, A. Movaghar, and F. Koroupi, "Investigating intrusion detection systems in MANET and comparing IDSs for detecting misbehaving nodes," in *Proceedings of World Academy of Science, Engineering and Technology*, vol. 34, October 2008.

[6] E. J. Caballero, "Vulnerabilities of intrusion detection systems in mobile ad-hoc networks—the routing problem," in *TKK T-110.5290 Seminar on Network Security*, 2006.

[7] T. Roosta, S. Shieh, and S. Sastry, "Taxonomy of security attacks in sensor networks," in *Proceedings of the 1st IEEE International Conference on System Integration and Reliability Improvements*, vol. 1, pp. 529–536, Hanoi, Vietnam, 2006.

[8] S. Khan, N. Mast, and J. Loo, "Denial of service attacks and mitigation techniques in IEEE 802.11 Wireless mesh networks," *Information*, vol. 12, pp. 1–8, 2009.

[9] S. Khan and J. Loo, "Cross layer secure and resource-aware on-demand routing protocol for hybrid wireless mesh networks," *Wireless Personal Communications*, vol. 62, no. 1, pp. 201–214, 2010.

[10] S. Khan, N. Mast, K.-K. Loo, and A. Silahuddin, "Passive security threats and consequences in IEEE 802.11 wireless mesh networks," *International Journal of Digital Content Technology and Its Applications*, vol. 2, no. 3, pp. 4–8, 2008.

[11] E. Shi and A. Perrig, "Designing secure sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 38–43, 2004.

[12] G. Padmavathi and D. Shanmugapriya, "A survey of attacks, security mechanisms and challenges in wireless sensor networks," *International Journal of Computer Science and Information Security*, vol. 4, no. 2, 2009.

[13] S. Mohammadi and H. Jadidoleslamy, "A comparison of physical attacks on wireless sensor networks," *International Journal of Peer to Peer Networks*, vol. 2, no. 2, pp. 24–42, 2011.

[14] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[15] M. Hussaini, H. Bello-Salau, A. Salami, F. Anwar, A. Abdalla, and M. Islam, "Enhanced clustering routing protocol for power-efficient gathering in wireless sensor network," *International Journal of Communication Networks and Information Security*, vol. 4, pp. 18–28, 2012.

[16] A. Popescu, G. Tudorache, B. Peng, and A. Kemp, "Surveying position based routing protocols for wireless sensor and ad-hoc networks," *International Journal of Communication Networks and Information Security*, vol. 4, pp. 41–67, 2012.

[17] O. Fdili, Y. Fakhri, and D. Aboutajdine, "Impact of queue buffer size awareness on single and multi service real-time routing protocols for WSNs," *International Journal of Communication Networks and Information Security*, vol. 4, pp. 104–111, 2012.

[18] J. Sen, "A survey on wireless sensor network security," *International Journal of Communication Networks and Information Security*, vol. 1, pp. 55–78, 2009.

[19] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[20] S. Khan, K.-K. Loo, N. Mast, and T. Naeem, "SRPM: secure routing protocol for IEEE 802.11 infrastructure based wireless mesh networks," *Journal of Network and Systems Management*, vol. 18, no. 2, pp. 190–209, 2010.

[21] S. Khan, N. A. Alrajeh, and K.-K. Loo, "Secure route selection in wireless mesh networks," *Journal of Computer Networks*, vol. 56, no. 2, pp. 491–503, 2012.

[22] D. Djenouri, L. Khelladi, and N. Badache, "A survey of security issues in mobile ad hoc and sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 4, 2005.

[23] M. S. Siddiqui and S. H. Choong, "Security issues in wireless mesh networks," in *Proceedings of the International Conference on Multimedia and Ubiquitous Engineering (MUE '07)*, pp. 717–722, April 2007.

[24] S. Khan and K.-K. Loo, "Real-time cross-layer design for a large-scale flood detection and attack trace-back mechanism in IEEE 802.11 wireless mesh networks," *Network Security*, vol. 2009, no. 5, pp. 9–16, 2009.

[25] I. Krontiris, T. Dimitriou, and F. C. Freiling, "Towards intrusion detection in wireless sensor networks," in *Proceedings of the 13th European Wireless Conference*, Paris, France, April 2007.

[26] I. Krontiris, T. Dimitriou, T. Giannetsos, and M. Mpasoukos, "Intrusion detection of Sinkhole attacks in wireless sensor

networks," in *Algorithmic Aspects of Wireless Sensor Networks ALGOSENSORS*, vol. 4837 of *Lecture Notes in Computer Science*, pp. 150–161, Springer, 2008.

[27] H. Jadidoleslamy, "A hierarchical intrusion detection architecture for wireless sensor networks," *International Journal of Network Security & Its Applications*, vol. 3, no. 5, 2011.

[28] A. Abraham, C. Grosan, and C. Martin-Vide, "Evolutionary design of intrusion detection programs," *International Journal of Network Security*, vol. 4, no. 3, pp. 328–339, 2007.

[29] A. Abraham, R. Jain, J. Thomas, and S. Y. Han, "D-SCIDS: distributed soft computing intrusion detection system," *Journal of Network and Computer Applications*, vol. 30, no. 1, pp. 81–98, 2007.

[30] A. P. R. Da Silva, A. A. F. Loureiro, M. H. T. Martins, L. B. Ruiz, B. P. S. Rocha, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Quality of Service and Security in Wireless and Mobile Networks (Q2SWinet '05)*, pp. 16–23, Montreal, Canada, October 2005.

[31] R. Roman, J. Zhou, and J. Lopez, "Applying intrusion detection systems to wireless sensor networks," in *Proceedings of the 3rd IEEE Consumer Communications and Networking Conference (CCNC '06)*, pp. 640–644, January 2006.

[32] S. Banerjee, C. Grosan, and A. Abraham, "IDEAS: Intrusion detection based on emotional ants for sensors," in *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA '05)*, pp. 344–349, September 2005.

[33] M. S. Islam and S. A. Rahman, "Anomaly intrusion detection system in wireless sensor networks: security threats and existing approaches," *International Journal of Advanced Sciences and Technology*, vol. 36, pp. 1–8, 2011.

[34] Y. Y. Li and L. E. Parker, "Intruder detection using a wireless sensor network with an intelligent mobile robot response," in *IEEE Conference Southeastcon*, pp. 37–42, April 2008.

[35] V. Bhuse and A. Gupta, "Anomaly intrusion detection in wireless sensor networks," *Journal of High Speed Networks*, vol. 15, no. 1, pp. 33–51, 2006.

[36] C. E. Loo, M. Y. Ng, C. Leckie, and M. Palaniswami, "Intrusion detection for routing attacks in sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2, no. 4, pp. 313–332, 2006.

[37] H. Deng, Q. A. Zeng, and D. P. Agrawal, "SVM-based intrusion detection system for wireless ad hoc networks," in *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC '03)*, pp. 2147–2151, October 2003.

[38] Y. A. Huang, W. Fan, W. Lee, and P. S. Yu, "Cross-feature analysis for detecting ad-hoc routing anomalies," in *Proceedings of the 23th IEEE International Conference on Distributed Computing Systems*, pp. 478–487, May 2003.

[39] I. Onat and A. Miri, "An intrusion detection system for wireless sensor networks," in *Proceedings of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '2005)*, pp. 253–259, August 2005.

[40] M. S. I. Mamun and A. F. M. Sultanul Kabir, "Hierarchical design based intrusion detection system for wireless ad hoc sensor network," *International Journal of Network Security & Its Applications*, vol. 2, no. 3, 2010.

[41] H. Sedjelmaci and M. Feham, "Novel hybrid intrusion detection system for clustered wireless sensor network," *International Journal of Network Security & Its Applications*, vol. 3, no. 4, 2011.

[42] R. Bhatnagar and U. Shankar, "The proposal of hybrid intrusion detection for defence of sync flood attack in wireless sensor network," *International Journal of Computer Science & Engineering Survey*, vol. 3, no. 2, pp. 31–38, 2012.

[43] K. Q. Yan, S. C. Wang, and C. W. Liu, "A hybrid intrusion detection system of cluster-based wireless sensor networks," in *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS '09)*, Hong Kong, 2009.

[44] K. Q. Yan, S. C. Wang, S. S. Wang, and C. W. Liu, "Hybrid Intrusion Detection System for enhancing the security of a cluster-based Wireless Sensor Network," in *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT '10)*, pp. 114–118, Chengdu, China, July 2010.

[45] T. H. Hai, F. Khan, and E. N. Huh, "Hybrid intrusion detection system for wireless sensor networks," in *Computational Science and Its Applications—ICCSA 2007*, vol. 4706 of *Lecture Notes in Computer Science*, pp. 383–396, Springer, Berlin, Germany, 2007.

[46] S. Khan, K.-K. Loo, and Z. U. Din, "Cross layer design for routing and security in multi-hop wireless networks," *International Journal of Information Assurance and Security*, vol. 4, no. 2, pp. 170–173, 2009.

[47] D. E. Boubiche and A. Bilami, "Cross layer intrusion detection system for wireless sensor network," *International Journal of Network Security & Its Applications*, vol. 4, no. 2, 2012.

[48] M. Xiao, X. Wang, and G. Yang, "Cross-layer design for the security of wireless sensor networks," in *Proceedings of the 6th World Congress on Intelligent Control and Automation (WCICA '06)*, pp. 104–108, Dalian, China, June 2006.