Design of sinkhole node detection mechanism for hierarchical wireless sensor networks

Article *in* Security and Communication Networks · October 2016

DOI: 10.1002/sec.1652

CITATIONS
44

READS
496

4 authors:

Mohammad Wazid
Graphic Era University
88 PUBLICATIONS **3,662** CITATIONS
SEE PROFILE

Ashok Kumar Das
International Institute of Information Technology, Hyderabad
277 PUBLICATIONS **8,832** CITATIONS
SEE PROFILE

Saru Kumari
Chaudhary Charan Singh University
240 PUBLICATIONS **6,485** CITATIONS
SEE PROFILE

Khurram Khan
Dadabhoy Institute of Higher Education
394 PUBLICATIONS **12,250** CITATIONS
SEE PROFILE

SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2016; **9**:4596–4614
Published online 17 October 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.1652

RESEARCH ARTICLE

Design of sinkhole node detection mechanism for hierarchical wireless sensor networks

Mohammad Wazid[1], Ashok Kumar Das[1]*, Saru Kumari[2] and Muhammad Khurram Khan[3]

[1] Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India
[2] Department of Mathematics, Ch. Charan Singh University, Meerut 250 005, Uttar Pradesh, India
[3] Center of Excellence in Information Assurance,King Saud University, Riyadh, Kingdom of Saudi Arabia

ABSTRACT

Wireless sensor networks (WSNs) have several applications ranging from the civilian to military applications. WSNs are prone to various hole attacks, such as sinkhole, wormhole, blackhole, and greyhole. Among these hole attacks, the sinkhole attack is the malignant one. A sinkhole attack allows a malicious node, called the sinkhole node, advertises a best possible path to the base station (BS). This misguides its neighbors to utilize that path more frequently. The sinkhole node has the opportunity to tamper with the data, and it also performs the modifications in messages or it drops messages or it produces unnecessary delay before forwarding them to the BS. On the basis of these malicious acts that are performed by a sinkhole attacker node, we consider three types of malicious nodes in a WSN: sinkhole message modification node (SMD), sinkhole message dropping node (SDP), and sinkhole message delay node (SDL). None of the existing techniques in the literature is capable to handle all three types of nodes at a time. This paper presents a new detection scheme for the detection of different types of sinkhole nodes for a hierarchical wireless sensor network (HWSN). To the best of our knowledge, this is the first attempt to design such a detection scheme in HWSNs which can detect SMD, SDP, and SDL nodes. In our approach, the entire HWSN is divided into several disjoint clusters, and each cluster has a powerful high-end sensor node (called a cluster head), which is responsible for the detection of different sinkhole attacker nodes if present in that cluster. We simulate our scheme using the widely-accepted NS2 simulator for measurement of various network parameters. The proposed scheme achieves around 95% detection rate and 1.25% false positive rate. These factors are significantly better than the previous related schemes. Furthermore, the computation and communication efficiency is achieved in our scheme. As a result, our scheme seems suitable for the sensitive critical applications, such as military applications. Copyright © 2016 John Wiley & Sons, Ltd.

KEYWORDS
Hierarchical wireless sensor networks, sinkhole attack, authentication, key management, NS2, security.

***Correspondence**
Ashok Kumar Das, Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India.
E-mail: iitkgp.akdas@gmail.com

1. INTRODUCTION

Wireless sensor networks (WSNs) consist of many small sized, inexpensive, and computable tiny devices which are called as sensor nodes. They have limited memory, computation power. and battery backup. Sensor nodes gather the sensing information such as pressure, temperature, concentration of hazardous gases in the environment, etc. from their surrounding and then send the sensed and processed information to the nearby base station(s) (BS) via a multihop wireless communication path for further computations and processing of the information.

Figure 1 depicts a typical scenario of a hierarchical wireless sensor network (HWSN), it maintains a hierarchy among the deployed nodes for example BS, cluster heads, and sensor nodes on the basis of their capabilities. Sensor nodes are generic wireless devices which have limited capabilities such as limited battery backup, memory size, and data processing and have short radio transmission range. A HWSN is divided into several clusters each cluster has some sensor nodes which can communicate among each other also their respective cluster head (*CH*) node. Usually, the *CHs* are resource-rich nodes than the resource-constrained sensor nodes having large memory size, high

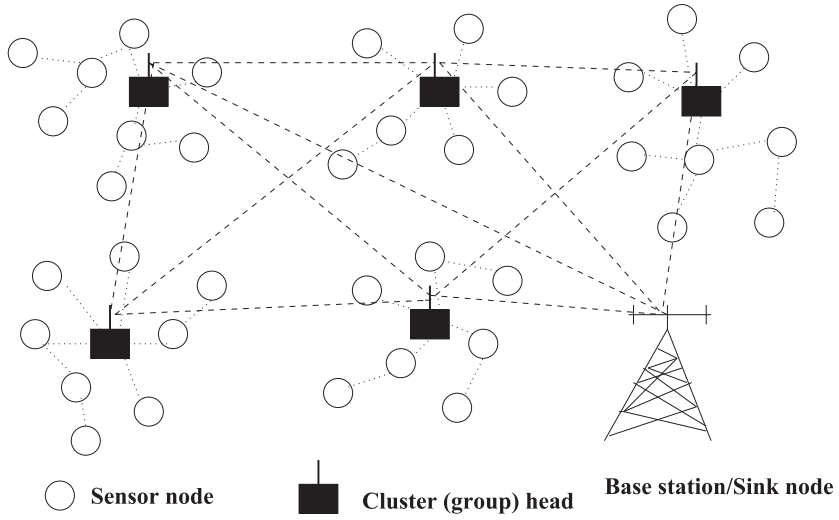

Figure 1. A hierarchical wireless sensor network (HWSN) architecture (Source: [1]).

battery backup, good data processing speed, and a powerful antenna. The *CHs* can execute complicated numerical operations very fast as compared with sensor nodes. The *CHs* either directly communicate among each other or via their neighbor *CHs*, and relay data between their members and the BS. A BS (sink node or gateway node) is the gateway to another network, which is treated as a powerful data processing/storage center and also an access point for human interface. The BS sends periodically the queries to the sensor nodes in order to collect the sensing gathered information, and then performs costly operations on those data on behalf of sensors and also manages the whole network. The BS is the trusted component in the network, and it is assumed that it will not be compromised by an adversary, otherwise the entire WSN will be compromised by the attacker. Sensor nodes are deployed around one hop or more hops neighborhood of their corresponding *CH*. All sensor nodes and *CHs* are reached by the BS in WSN as the BS is the most powerful node. The deployment location of the BS depends on one application to another application, and thus, BS can be located either in the center or at a corner of the network. There are three types of data flows in HWSN: (i) pairwise (unicast) among sensor nodes, (ii) group-wise (multicast) within a cluster of sensor nodes, and (iii) network-wise (broadcast) from BS to sensor nodes and *CHs*. The seminal survey paper can also be considered for more details [2]. As specified in [1,3], the *CHs* are high-end sensor nodes (for example, PDAs) and the sensor nodes are low-end sensor nodes (for example, micaZ motes).

WSNs are used in many civil and military applications. Security is a major concern for a WSN, which is deployed for some military applications. An attacker node, which is deployed by an enemy to capture the exchanging confidential information among the sensor nodes. There is a great need for an intrusion detection system (IDS) in WSNs to detect the existence of malicious attacker nodes. So intrusion detection schemes have received a great deal of attention because they support different applications such as military surveillance and environmental monitoring [4,5].

The hole attacks such as sinkhole, wormhole, blackhole, greyhole, etc. are some of the dangerous attacks, which can happen in a WSN. In the existence of these attacks, the information do not reach the destination within time, sometime the information are lost or modified, and also causes large energy expenditure [6–10].

A wormhole attack allows an attacker to form a tunnel between two distant locations in WSN, and using that tunnel, the packets are sent through an in-band channel or out-of-band channel [11]. Thus, a wormhole tunnel is formed by a pair of attackers. It permits two distant nodes a misapprehension that they are close to each other; however, in practice it is not so. The existing wormhole can then attract and bypass a large amount of the network traffic in WSN. Therefore, the wormhole node can easily capture the traffic in WSN and perform several manipulations on the traffic. As a result, a variety of attacks including the sniffing, modification, and dropping can be launched by a wormhole attacker node.

In a blackhole attack, an attacker has the opportunity to capture and re-program a set of sensor nodes to block the packets that they receive instead of forwarding those packets towards the BS in WSN [10]. Therefore, the information which pass through the blackhole region are compromised by the blackhole attacker. This attack can undermine the network effectiveness by partitioning WSN such that the important information never reach to the BS.

The grayhole attack is the variation of the blackhole attack in which the malicious attacker node drops the packets selectively or with certain probability. The attacker node drops packets from a particular node and forwards the packets of the other nodes. It may also drop on the basis of traffic type, for example, it may drop all the user datagram protocol (UDP) packets while forwarding all the

transmission control protocol (TCP) packets [12], [13], [14]. The presence of the wormhole, blackhole, and grayhole nodes in WSN can affect various network performance parameters, for example, throughput and EED.

We explain the detailed mechanism of sinkhole attack below as it is our main focus in this paper.

1.1. Sinkhole attack

In a sinkhole attack, a sinkhole attacker node first advertises a best possible route (with less hop-distance route) to the destination (BS) to attract its neighbors so that they may fall into this attraction to utilize the advertised route more frequently. The neighbors can then forward their traffic through the efficient advertised route declared by the sinkhole attacker node. The route can also captivates other nodes apart from the neighbor nodes of the sinkhole attacker node, which are closer to the sinkhole than to BS. So the attacker node has the opportunity to tamper with the data, damage the regular network operations, or conduct other serious threats [9]. A scenario of this type of sinkhole attack is illustrated in Figure 2.

The sinkhole attack can be mounted using the wormhole attack, wherein a malicious node first captures the packets from its neighbors, and then it uses a secret wormhole tunnel in order to forward the packets to another colluded node in WSN, which is responsible to eventually deliver the packets to the BS. Note that the two ends of the wormhole tunnel can be at a longer distance as compared with other routes. However, it prevents the source from discovering other routes, which are greater than two hops away from the BS. A scenario of such attack is illustrated in Figure 3.

If a sinkhole attacker node is deployed successfully, there will be three possibilities: messages may be lost (dropped by the attacker node), messages may be delayed, or messages may be modified [8], [9], [15], [16]. On the basis of these three observations, three types of sinkhole attacker nodes are possible:

- Sinkhole message modification nodes (SMD): Sinkhole attacker nodes modify the messages before forwarding them to the next node.

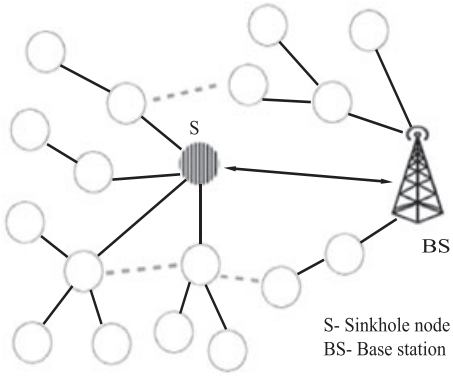

Figure 2. An illustration of sinkhole attack scenario (Source: [9]).

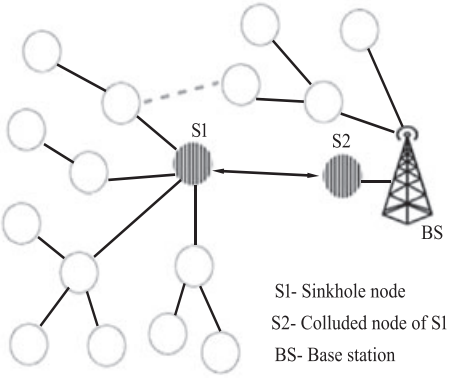

Figure 3. An illustration of sinkhole attack scenario using wormhole tunnel (Source: [9]).

- Sinkhole message dropping nodes (SDP): Sinkhole attacker nodes drop the messages, even sometimes selectively.
- Sinkhole message delay nodes (SDL): Sinkhole attacker nodes cause delay in forwarding the messages.

In the presence of sinkhole attacker nodes, messages may be modified or delayed or dropped. This causes serious threats to the functioning of WSN, where the information cannot reach to the BS in time, and other network parameters are also affected. In this paper, we propose an efficient new detection scheme for different types of sinkhole attacker nodes (SMD, SDP, and SDL) in HWSNs. To the best of our knowledge, this is the first attempt to design such sinkhole nodes detection approach in HWSNs, which can effectively detect SMD, SDP, and SDL attacker nodes. The proposed detection scheme is divided into two phases. In phase 1, we detect the existence of sinkhole attacker nodes, and in phase 2, we identify their types (SMD, SDP, or SDL).

1.2. Motivation

In most applications, WSNs are deployed in an unattended environment [17], [18], [19], [20], [21], [22], [23], [24]. Thus, the sensor nodes inside WSNs can be easily captured by enemies, which cause information loss along with large energy expenditure. Therefore, securing WSNs is very important in designing of a wireless sensor network. WSN is prone to various attacks and the confidential information can be leaked or altered. A sinkhole attacker node advertises a best possible route to the BS which misguides its neighbors in order to use that route more frequently. Thus, in the presence of sinkhole attack, messages may be lost, modified, or dropped by the attacker nodes that causes serious threat to the network. The exiting schemes for defending sinkhole attack are not efficient enough as they have some limitations which are discussed in Table I. Therefore, we feel that there is a great need for an efficient detection scheme for the detection of different

Table I. Summary of technique used and limitations/drawbacks of the existing protocols.

Scheme	Technique used	Limitations/drawbacks
Du *et al.* [37] (2007)	Two Tier Secure Routing (TTSR)	very low PDR with less number of L-sensors
Wang *et al.* [5] (2008)	Single and multi sensing detection	low *DR*
Krontiris *et al.* [33] (2008)	Cooperative detection by using intersection of neighbor lists	low *DR*
Wang *et al.* [27] (2011)	IHIDS for the sink, HIDS for CH and misuse IDS approaches	low *DR* and high computational cost
Wang *et al.* [4] (2013)	Gaussian and uniformly distributed WSN	low *DR* with less number of nodes
Salehi *et al.* [44] (2013)	Grouping of suspected nodes and network information flow based detection	high *FPR*
Hamedheidari *et al.* [7] (2013)	Mobile agent based detection	high network overhead
Shafiei *et al.* [9] (2014)	Energy holes estimation by using geostatistical hazard model	energy expenditure maps can create problem in network congestion areas that further affects *DR* and *FPR*
Zhang *et al.* [30] (2014)	Redundancy mechanism	low *DR*

Note: DR : detection rate; *FPR* : false positive rate; *PDR* : packet delivery ratio

types sinkhole attacker nodes. We propose a new cluster based detection scheme for sinkhole attacker nodes, which has the capability to detect and defend different types of sinkhole attacker nodes effectively in HWSNs.

1.3. Threat model

Our scheme utilizes the well-known Dolev-Yao threat model [25], in which any two communicating nodes (parties) can communicate over an insecure channel [26]. We have used same type of threat model in our scheme, where the channel is insecure, and the end-points (sensor nodes/CHs) are not in general trustworthy. In the presence of the sinkhole attacker nodes in WSN, the messages may be lost, modified, or dropped, and this causes serious threat to the performance to the network such as reduction in packet delivery ratio (PDR) and throughput, and high EED. We further assume that an attacker can physically capture some sensor nodes or CHs in the network and using the information stored in those nodes, the attacker can directly deploy some malicious nodes in the network, which can act as sinkhole attacker nodes.

1.4. Our contributions

The contributions made in this paper are listed below:

- We propose a new cluster based scheme in which powerful *CH* nodes detect the different types of sinkhole attacker nodes.
- Our scheme is better than the other existing schemes as it detects different types of sinkhole attacker nodes at the same time.
- Our scheme is secure against sinkhole attacker nodes, which is shown through the analytical and simulation results using the widely-accepted NS2 simulator.
- Finally, our scheme performs better as compared with the related existing schemes in WSNs.

1.5. Organization of the paper

The rest of the paper is organized as follows. Section 2 reviews the related existing sinkhole attack detection schemes in WSNs. In Section 3, we present our proposed cluster based scheme for detection of sinkhole nodes in WSN. In Section 4, we provide mathematical analysis of our scheme for the traffic model. In Section 5, through the security analysis, we show that our scheme has the ability to resist different types of sinkhole attacker nodes. We perform the simulation of our scheme using the widely-accepted NS2 tool in Section 6, and the rigorous simulation results of the proposed scheme are also provided in this section. The performance of our scheme is compared with related existing schemes in Section 7 and the paper is concluded in Section 8.

2. LITERATURE SURVEY

Ngai *et al.* [8] proposed a lightweight technique for detecting sinkhole attack. In their technique, the attackers are detected by observing the network flow information. A many-to-one communication model is used in which the routes are established based on route advertisements received. The scheme has less communication and computation overheads. But the success rate of the scheme is less in case of high drop rate.

Wang *et al.* [27] proposed an integrated intrusion detection system (IIDS) for a cluster based WSN. The proposed system is capable to resist the attacks by doing the real-time analysis of network data. The proposed system consists of three types of IDSs such as intelligent hybrid intrusion detection system (IHIDS), hybrid intrusion detection system (HIDS), and misuse intrusion detection system. The detection is performed using anomaly and misuse detection modules which gives high detection rate (DR) with low false positive (FP) rate.

Hamedheidari *et al.* [7] proposed a mobile agent based defensive mechanism against sinkhole attack. They used mobile agents to aware all sensor nodes from its neighbors through a three-step negotiation, which makes them not to listen the traffics generated by the malicious sinkhole attacker nodes. In the performance evaluation, the scheme is evaluated in terms of mobile agent energy consumption, packet loss rate, throughput, etc. The use of mobile agents produces the network overhead, which is a drawback for a WSN. Fessant *et al.* [15] proposed technique to describe the impact of selective-forwarding attacks in tree-based routing protocols. The proposed protocol is effective and improves the resilience of WSN against sinkhole attacks.

Zhu *et al.* [28] proposed a technique to detect node replication attack in which adversaries prepare their own low-cost sensor nodes and then deploy them in the deployment field that causes the network to accept them as legitimate nodes. To prepare the clone of the sensor node, an adversary can physically capture the sensor node and extract all its confidential information such as keys used for the communication, its identification number (ID), etc. and then reproduce their own nodes by using that extracted information and deploy them in the network at strategic positions. Shafiei *et al.* [9] proposed a method to identify energy holes (sinkholes). Sinkhole attacker nodes are detected using a centralized model. A lightweight mitigation method is also provided to eliminate sinkhole attacker nodes.

Rajasegarar *et al.* [29] proposed a distributed hyperspherical cluster based algorithm to identify anomalies in WSN. The implementation of the proposed schemes on a real WSN testbed is also performed. The distributed hyperspherical cluster-based scheme has better detection accuracy with less communication overhead as compared with the centralized scheme in which all sensor nodes communicate to a central node for processing. Zhang *et al.* [30] proposed a redundancy mechanism to prevent sinkhole attack. In this technique, messages are sent to the suspicious nodes through multiple paths. The sinkhole attacker nodes are identified on the basis of replied messages received from the suspicious nodes. The simulation are performed in NS2 to test the effectiveness of the scheme. However, their proposed scheme has low DR. Sreelaja *et al.* [31] proposed an ant colony optimization attack detection (ACO-AD) algorithm to detect sinkhole attacker nodes. In this scheme, nodes generate an alert if they identify any sinkhole in the network. A voting based method is used to identify sinkhole attacker nodes. Their proposed technique identifies the anomalous connections without generating FP and with minimum storage memory use of sensor nodes.

Nahas *et al.* [32] proposed a novel routing approach to protect a WSN against the wormhole and sinkhole attacks, which is called the Secure-Path Routing (SPR). Their method uses expected path risk as a parameter in routing which are further used in routing to reduce the traffic flow over the nodes which are vulnerable to holes attacks. But the problem with the selection of low risk routes may lead to the choice of routes which can consume large energy. Thus, they implemented an algorithm that balances the risk with other path selection parameter such as energy consumption. They also evaluated the trade-off between security and energy consumption. During the experimentation, it has been observed that the proposed technique is quite effective as it increases the traffic flow over legitimate routes and its impact on the network lifetime is negligible.

Krontiris *et al.* [33] proposed an intrusion detection system to protect WSN against sinkhole attack. Some rules are designed and embedded in the IDS system for the successful detection of sinkhole attack. However, their proposed scheme has low DR. Garofalo *et al.* [34] proposed a decision tree classification based technique for the detection of sinkhole attack. There is trade-off between high DR and used energy in detection process. So a light weight detection technique is implemented on motes in order to save the energy. Sinkhole attack dataset has been created and utilized to evaluate the effectiveness of the proposed scheme.

Giruka *et al.* [35] surveyed the state of art approaches in securing WSNs. Several security techniques of WSN were reviewed. Mainly focused on authentication, key management and distribution and secure routing techniques available for intrusion detection in WSNs. Hai *et al.* [36] proposed a lightweight IDS for cluster-based WSN. An algorithm was proposed to minimize the triggered intrusion modules in the network by using an over-hearing technique to reduce the sending alert packets. The proposed technique is capable to detect most of the routing attacks in WSN. During the experimentation, it has been observed that their technique requires less energy consumption as compared with the other techniques. However, their technique has high FP rate up to 10% in some cases.

Du *et al.* [37] proposed a secure and efficient routing protocol for heterogeneous sensor networks. Their scheme specifically utilizes the powerful high-end sensors. During the experimentation, it has been observed that their secure routing protocol achieves better routing performance than the existing directed diffusion technique. The delivery ratio of the proposed scheme decreases with increase in failure nodes. The delivery ratio is good with high number of L-sensors which decreases drastically in case of less number of L-sensors. Dallas *et al.* [38] proposed a method for the detection of sinkhole attack, or the other attack which misleads traffic by considering the cost of an attack route. They monitor the hop-count parameter in order to detect sinkhole attack. Their scheme is computationally efficient for

detecting the abnormal route advertisements that are used to perform sinkhole attack. Roy *et al.* [39] implemented a dynamic trust management system (DTMS) that counters two severe attacks (sinkhole and blackhole attacks) in WSN. Their technique ensures that network architectures do not require redefinition for every specific attack. It can handle both attacks at the same time. The drawback with such schemes is that they require high computational cost which creates energy consumption issues in low powered sensing devices.

Krontiris *et al.* [40] investigated the impact of severe routing attack, for example, sinkhole attack in a WSN. They proposed a technique to countermeasure against the sinkhole attack. Papadimitriou *et al.* [16] introduced two cryptographic techniques to secure a WSN against sinkhole attack. The objective of the proposed technique is to provide continuous protection against the sinkhole attacker nodes rather than its detection. Their proposed cryptographic protocols are effective against the sinkhole attack as they successfully protect the network against sinkhole attacker nodes. Chen *et al.* [41] proposed a technique to protect the large scale WSNs against the sinkhole attack. The detection problem is formulated as a change-point detection problem in which they monitor the CPU usage of each sensor node and predict the normal or abnormal behavior on the basis of CPU usage. Their proposed technique is capable to differentiate among normal and abnormal sensor nodes (attacker nodes). However, CPU usage is also more in case of other attacker nodes, for example, blackhole, wormhole nodes, or in some other attacks. Therefore, it is very difficult to confirm that the suspicious node is sinkhole attacker node by using the proposed scheme.

Traditional cryptographic schemes used in the development of trust-aware routing protocols do not effectively address the problems associated with multihop routing. The attacks such as sybil, sinkhole, and wormhole are possible in WSN. So to secure WSNs against adversaries misdirecting the multihop routing, Zhan *et al.* [42] proposed a trust-aware routing framework (TARF) for dynamic WSNs. TARF is efficient as it provides trustworthy and energy-efficient routes without requiring time synchronization and geographic information of sensor nodes. The TARF module is implemented and demonstrated on TinyOS platform. Simulation results prove its effectiveness against the various routing attacks. Qi *et al.* [43] used MultiHopLQI routing protocols to protect the WSN against sinkhole attack. The MultiHopLQI routing protocol adopts link quality indicator (LQI) that indicates the last packet as the criterion for parent selection. The aim of MultiHopLQI mechanism is to ensure that a message reaches to the BS within time and in an accurate form.

Salehi *et al.* [44] proposed a detection mechanism for sinkhole attack. The suggested algorithm first identifies a group of suspected nodes and then the sinkhole attacker nodes are confirmed on the basis of network flow information. Simulations are performed to check the effectiveness of the proposed scheme. Their technique has low DR and high FP rate. Sharmila *et al.* [45] proposed a message digest algorithm based technique to detect sinkhole attack in WSNs. The proposed technique ensures the integrity of the transferred messages using a trustable path.

In WSN, an intruder can be a moving object deployed by the enemy in the battlefield. With uniformly deployed wireless sensor nodes, the detection probability is the same at any point. But the detection probability is application specific and can also vary as per the location. The sensor nodes deployed using the Gaussian distribution method provides differentiated detection capabilities at different locations in WSN. The problem of intrusion detection in a Gaussian-distributed WSN is analyzed by Wang *et al.* [4]. Two types of detection scenarios such as single-sensing detection and multiple-sensing detection are considered. The performance of Gaussian-distributed WSNs is also compared with the performance of uniformly distributed WSNs. In WSN, the intrusion detection mechanism to detect the presence of static or moving attackers is also required. Wang *et al.* [5] discussed this issue to characterize WSN parameters such as node density, sensing range, etc. in terms of a desirable detection probability. The issue is considered according to the homogeneous and heterogeneous WSN models. The detection probabilities for both single-sensing and multiple-sensing methods were also computed. The simulation results proved the effectiveness of the scheme for both homogeneous and heterogeneous WSN models.

Finally, the summary of the techniques used and limitations/ drawbacks of the existing state-of-art protocols is provided in Table I.

3. THE PROPOSED SCHEME

In this section, we first discuss the various notations and network model used in our scheme. We then provide the high-level description and various message formats used in describing our scheme. Finally, we discuss, in detail, the proposed scheme.

3.1. Notations

We use the notations listed in Table II for describing our scheme.

3.2. Network model

In the proposed scheme, HWSN is divided into various disjoint clusters. Each cluster has is a high-end sensor node (H-sensor), which is a powerful node and acts as a *CH* as shown in Figure 1. For example, *CH* can be PDA [1], [3], [46]. Each cluster contains a number of the low-end sensor nodes (called the regular sensor nodes or L-sensors) as its members, which are extremely resource-constrained (for example, L-sensors can be MICAz motes [1]).

The L-sensors are deployed randomly in their respective clusters. The L-sensors first locate their neighbors

Table II. Notations used in this paper.

Symbol	Description
CH_j	j^{th} cluster head
S_i	i^{th} sensor node in a cluster
SMD	sinkhole message modification node
SDP	sinkhole message dropping node
SDL	sinkhole message delay node
ID_{S_i}	identity of S_i
ID_{CH_j}	identity of CH_j
SK_{CH_j,S_i}	secret key shared between CH_j and a member (sensor) S_i in a cluster
REN_{S_i}	remaining energy at sensor node S_i
MSG_r	message received at the cluster head
MSG_o	original message sent by the source node
T_{MSG_r}	receiving time of the message
T_{MSG_o}	actual receiving time of the message under normal flow
T_{con}	congestion delay
M_{sdq}	status-cum-data query message
M_{sr}	status reply message
M_d	data message
sdq_{rq}	information field in the status-cum-data query message M_{sdq}
s_{rp}	information field in the status reply message M_{sr}
$data$	data field in data message M_d
WT	waiting time at a cluster head CH_j
WT_θ	waiting time threshold at a cluster head CH_j
PDR_n, PDR_a, PDR_s	packet delivery ratio under normal flow, under attack and under our proposed scheme
DR	detection rate (sometimes also called as TPR)
FPR	false positive rate
Δ_n, Δ_a, Δ_s	end-to-end delay (ms) under normal flow, under attack and under our proposed scheme
TH_n, TH_a, TH_s	throughput ($kbps$) under normal flow, under attack and under our proposed scheme
M_d	data packet sent by sensor node
SA	sinkhole attacker node
$\|M_{d'}\|$	number of actual data packets received at cluster head CH_j
$\|M_{dsdpa}\|$	number of data packets dropped by SDP nodes.
$\|M_{dsdpa'}\|$	number of data packets dropped by actual SDP nodes (true positives)
$\|M_{d_1}\|$	number of data packets dropped by SDP nodes (false negatives)
$\|M_{dsdla}\|$	number of data packets delayed by SDL nodes
$\|M_{dsdla'}\|$	number of data packets delayed by actual SDL nodes (true positives)
$\|M_{d_2}\|$	number of data packets delayed by SDL nodes (false negatives)
M_{inf}	information message sent by each CH_j to all its members
T_{rec_i}	receiving time of a data packet i
T_{send_i}	sending time of a data packet i
p	total number of packets
pkt	data packet
$\|pkt\|$	data packet size
$h(\cdot)$	one-way collision-resistant cryptographic hash function
$HMAC$	hashed message authentication code
$A\|\|B$	data A concatenates with data B

within their communication ranges. For this purpose, each node needs to broadcast a HELLO message having its own identifier to other nodes in its communication range. After receiving other HELLO messages, each L-sensor prepares a list of its neighbor nodes [47]. Each *CH* in its cluster also finds its physical neighbors which are the L-sensors. Finally, each *CH* locates its neighbor *CH*s in its communication range.

The taks of a *CH* node is to detect the anomaly in the corresponding cluster in our scheme. Because of unattended environment, a *CH* can be physically captured or compromised by an adversary. Therefore, we assume that a compromised *CH* will be detected by the BS [48,49]. As a result, the deployment of a new *CH* is necessary to perform the detection of the sinkhole nodes.

For secure communication between neighboring cluster heads, between a *CH* and its sensor nodes (members), and also between sensor nodes in each cluster, we use the unconditionally secure deterministic key management scheme proposed by Das [46]. Let SK_{S_i,S_j}, SK_{CH_i,CH_j} and

SK_{CH_j,S_i} be the symmetric secret keys between two neighbor sensors S_i and S_j, between two cluster heads CH_i and CH_j, and between a cluster head CH_j and its neighbor member sensor node S_i in that cluster established using the deterministic key management scheme [46]. Thus, we assume that any two neighbor nodes in the network can securely communicate among each other using their corresponding established secret keys.

As the method mentioned in [50], we calculate the delay from L-sensor nodes to *CH*. Each packet sent by a node has a unique increasing sequence number. Let the receiving time of the packet i on node j with respect to the perfect clock be $t_r(i,j)$ and the transmitting time of the packet i on node j be $t_x(i,j)$. The transmitting or receiving time is the time just before the first byte of a packet is transmitted or received. Let O and D be the source and destination nodes for a path, respectively, $t_r(i,O)$ is then considered as the generation time of packet i on the source node. The EED of the packet i for a path is given by

$$t_d(i) = t_r(i,D) - t_r(i,O).$$

If the waiting time for packet i at node j on the path be $t_w(i,j)$, then $t_w(i,j) = t_x(i,j) - t_r(i,j)$. The waiting time $t_w(i,j)$ contains the backoff time on node to contend for the channel. The EED is then given by

$$t_d(i) = \sum_{j=1}^{n-1} t_w(i,j),$$

where n denotes the number of nodes in the path. Note that, in our scheme, O represents an L-sensor (regular sensor node), whereas D denotes *CH* node.

According to our threat model (described in Section 1.3), a sensor node in a cluster can be physically captured by an attacker. In general, the sensor nodes are not equipped with the tamper-resistant hardware. Thus, the attacker can extract all the sensitive information stored in that sensor's memory, such as its identity and secret key. The attacker can then store these extracted information and also the sinkhole attack functionality programs for SMD, SDP, and SDL in a fake sensor node, and then deploy in that cluster in WSN.

3.3. High-level description of the proposed scheme

Each *CH* has the information, such as the identity (ID_{S_i}) of each member (sensor node S_i) and their battery backup. According to the characteristics of sinkhole attack, the malicious sinkhole attacker node, advertises a best possible route towards the destination which misguides its neighbors in order to use that route more frequently. The malicious node thus gets an opportunity to tamper with the data, damage the regular network operations, or conduct other serious threats [9]. As discussed in Section 1, three types of sinkhole attacker nodes are possible, which are SMD, SDL, and SDP nodes. The detection of sinkhole attacker nodes is divided into two phases. In phase 1, we detect the existence of sinkhole attacker node by using Algorithm 1. It uses the parameters such as node identity ID_{S_i}, path information from source to destination containing hop count k_{S_i} and coefficient of the suspected node c_{S_i}, and remaining energy at the nodes REN_{S_i}. A node S_i is detected as a sinkhole attacker node if $k_{S_i} \leq c_{S_i}$ and $REN_{S_i} < REN_{S_{i_\theta}}$.

In phase 2 of the sinkhole attack detection, we identify the different types of sinkhole nodes such as SMD, SDL, and SDP by using Algorithm 2. The SMD nodes are identified by the CH_j by the hashed message authentication code (HMAC) procedure. Suppose for a sinkhole node S_i, a cluster head CH_j receives a message MSG_r that is not same as the original message MSG_o which was sent by a source node. There is a mismatch between the hash values of MSG_r and MSG_o. In that condition, the cluster head CH_j confirms that sinkhole node S_i is a SMD node. If a cluster head CH_j observes that the messages are delayed by some sinkhole attacker node S_i as the receiving time of a message T_{MSG_r} is greater than the actual receiving time T_{MSG_o}, in that situation CH_j checks for the other factors such as network congestion. If there is no congestion, sensor node S_i is detected as a SDL node. If a cluster head CH_j does not receive messages from sinkhole node S_i, it checks whether the node S_i is a SDP node or there is some other problem in the network such as node failure. CH_j sends status-cum-data query message M_{sdq} to node S_i, and the waiting time WT is incremented each time. If the waiting time expires, that is, $WT > WT_\theta$ where WT_θ is the waiting time threshold value, and CH_j does not receive response message (M_{sr}) and data message (M_d) from S_i, it is considered that node S_i fails. Otherwise, if CH_j receives response message (M_{sr}), but it does not receive data message (M_d), node S_i is detected as the SDP node.

In the anomaly alarm system phase, CH_j blacklists the detected sinkhole attacker node S_i and generates an alarm to aware the other cluster members that S_i is a sinkhole attacker node so that other members do not communicate with the malicious node S_i.

The high-level description of various phases related to our scheme is shown in Figure 4. These phases are summarized below:

- The network behavior analysis is performed by each cluster head *CH* in HWSN, and then *CH* checks the normal and abnormal behaviors of the network.
- First phase of detection (phase 1): The anomaly detection is performed by each *CH* using the sinkhole node existence algorithm provided in Algorithm 1.
- Second phase of detection (phase 2): If *CH* detects a sinkhole node, it runs the sinkhole node identification algorithm (Algorithm 2) to identify which types of sinkhole attacker nodes exist in the network.
- If *CH* detects the existence of sinkhole nodes and also identifies their types, it blacklists those malicious nodes and sends alarms to its cluster members (sensor nodes).

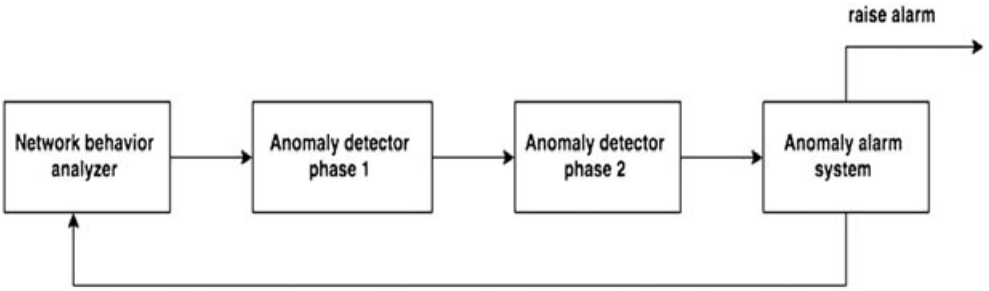

Figure 4. High-level description of proposed scheme.

3.4. Message formats

In our scheme, we use four different types of messages: (i) status-cum-data query message M_{sdq}, (ii) status reply message M_{sr}, (iii) data message $M_{d,}$ and (iv) information message M_{inf}. The format of each message is described below.

- *Status-cum-data query message* (M_{sdq})*:* The structure of this message is shown in Figure 5. This message is sent by a cluster head CH_j to its all members (sensors) S_i's in that cluster. It includes the following fields: the identity ID_{CH_j} of a cluster head CH_j, the identity ID_{S_i} of a sensor node (member) S_i of CH_j, the information field $sdq_{rq,}$ and the hashed message authentication code $HMAC_{msdq}$, where $HMAC_{msdq} = h(SK_{CH_j,S_i}||ID_{CH_j}||ID_{S_i}||sdq_{rq})$.
- *Status reply message* (M_{sr})*:* The structure of this message shown in Figure 6 has the following fields: ID_{S_i} is the identity of a sensor node S_i in a cluster, REN_{S_i} the remaining energy (battery) of S_i, s_{rp} the information field, and $HMAC_{msq} = h(SK_{CH_j,S_i}||ID_{S_i}||REN_{S_i}||s_{rp})$. This message is sent by a cluster member (sensor) S_i to its corresponding cluster head CH_j. In order to save energy, sensor nodes use multiple modes such as sleep, idle, and working [51], [52]. In the detection of SDPs, we consider two modes: idle and working as nodes cannot reply in sleeping state. As a result, s_{rp} field has two reply values: 0 means the idle mode and 1 means the working mode.

ID_{CH_j}	ID_{S_i}	sdq_{rq}	$HMAC_{msdq}$

Figure 5. Structure of status-cum-data query message (M_{sdq}).

ID_{S_i}	REN_{S_i}	s_{rp}	$HMAC_{msq}$

Figure 6. Structure of status reply message $M_{sr.}$

ID_{S_i}	REN_{S_i}	$data$	$HMAC_{md}$

Figure 7. Structure of data message $M_{d.}$

ID_{CH_j}	det_{inf}

Figure 8. Structure of Information message (M_{inf}).

- *Data message* (M_d) : The structure of this message shown in Figure 7 has the following fields: ID_{S_i} is the identity of a sensor node S_i in a cluster, REN_{S_i} the remaining energy (battery) of S_i, *data* the sensing information to be sent to its CH, and $HMAC_{md} = h(SK_{CH_j,S_i}||ID_{S_i}||REN_{S_i}||data)$. Note that *data* can be encrypted using the symmetric key SK_{CH_j,S_i}, if necessary.
- *Information message* (M_{inf}) : After the detection of sinkhole nodes, the cluster head CH_j sends the information message to alert its cluster members. The structure of the message shown in Figure 8 has the following fields: ID_{CH_j} is the identity of the cluster head CH_j and a detection information field det_{inf} that contains the information about the detected sinkhole nodes. This message is sent by a cluster head CH_j to its cluster members (sensors) except the blacklisted sinkhole attacker nodes in that cluster.

3.5. Description of the proposed scheme

In this section, we propose a scheme for the detection of different types of sinkhole attacker nodes such as SMD, SDP, and SDL nodes. Note that we have used the network model provided in Figure 1 (also discussed in Section 3.2) under which the proposed sinkhole node existence algorithm as well as sinkhole node identification algorithm works. The detection process has two phases: Phase 1 and Phase 2. In Phase 1, we identify the suspected sinkhole attacker nodes by using the sinkhole node existence algorithm. In Phase 2, we confirm which type of node, such as *SMD*, SDP, or *SDL* exists in the network by using sinkhole node identification algorithm. Both the algorithms are discussed below.

3.5.1. Sinkhole node existence algorithm.

The sinkhole node existence algorithm provided in Algorithm 1 is used to find out whether sinkhole attacker nodes exist in the network or not. It uses parameters such as node identification ID_{S_i}, path information from source to destination containing hop count k_{S_i}, and coefficient

Algorithm 1 Sinkhole node existence algorithm

for each cluster C_j in HWSN **do**
  CH_j sends status-cum-data query message (M_{sdq}) to its all cluster members (sensor nodes), S_i.
  After receiving M_{sdq}, each S_i recomputes $HMAC'_{msdq} = h(SK_{CH_j,S_i} \| ID_{CH_j} \| ID_{S_i} \| sdq_{rq})$ using the shared secret key SK_{CH_j,S_i} with CH_j.
  if ($HMAC'_{msdq} = HMAC_{msdq}$) **then**
    M_{sdq} is valid and S_i responses with status reply message $M_{sr} = \langle ID_{S_i}, REN_{S_i}, s_{rp}, HMAC_{msq} \rangle$ to CH_j using its current remaining energy REN_{S_i}.
    After receiving M_{sr}, CH_j recomputes $HMAC'_{msq} = h(SK_{CH_j,S_i} \| ID_{S_i} \| REN_{S_i} \| s_{rp})$ using the shared secret key SK_{CH_j,S_i} with S_i.
    if ($HMAC'_{msq} = HMAC_{msq}$) **then**
      M_{sr} is valid.
    end if
  end if
  Each S_i in C_j sends data message $M_d = \langle ID_{S_i}, REN_{S_i}, data, HMAC_{md} \rangle$, if any, to CH_j using its current remaining energy REN_{S_i}.
  After receiving M_d from S_i, CH_j recomputes $HMAC'_{md} = h(SK_{CH_j,S_i} \| ID_{S_i} \| REN_{S_i} \| data)$ using the shared secret key SK_{CH_j,S_i} with S_i.
  if ($HMAC'_{md} = HMAC_{md}$) **then**
    M_{md} is valid.
  end if
  On the basis of information provided by sensor nodes S_i, CH_j computes all paths $P = \cup_{i=1}^{n} P_i$ from itself to all destination cluster members.
  Check the coefficients of all intermediate nodes along the paths.
  if any intermediate node, say X has high coefficient value c_X **then**
    Compute X's hop count value k_X from CH_j.
    if ($k_X \leq c_X$) **then**
      Compute the remaining energy REN_X.
      if ($REN_X < REN_{X_\theta}$) **then**
        Node X is considered as a sinkhole node.
        Call sinkhole node identification algorithm, Algorithm 2.
      end if
    end if
  end if
end for

of the suspected node c_{S_i}, and remaining energy at the nodes REN_{S_i}.

The sinkhole node existence algorithm detects sinkhole nodes on the basis of hop count value k_{S_i} and path coefficient c_{S_i} of the suspected intermediate nodes S_i such that $k_{S_i} \leq c_{S_i}$ and also the remaining energy is less than the remaining energy threshold value for that node $REN_{S_i} < REN_{S_{i_\theta}}$. The exchanged messages in the sinkhole node existence algorithm are summarized in Figure 9.

Remark 1. Let the hop count value for a suspected intermediate node S_i be k_{S_i} and path coefficient be c_{S_i}. If the condition $k_{S_i} \leq c_{S_i}$ holds, the suspected intermediate node S_i is a sinkhole node.

Example 1. To validate remark 1, consider the scenarios given in Figures 10 and 11. There are two source nodes (S_1 and S_2) which send some information to destination node D (a cluster head node). The messages are communicated via two paths, say $P_1 = \langle S_1, P, Q, R, S, D \rangle$ and $P_2 = \langle S_2, V, X, Y, Z, D \rangle$ as shown in Figure 10.

Suppose there exists a sinkhole attacker node H in the network that advertises an efficient path to the destination D. This attracts the other neighbor nodes to select node H as the next hop. So the entire traffic flow starts through the

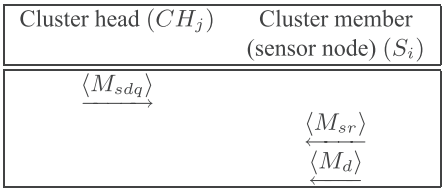

Figure 9. Exchanged messages in the sinkhole node existence algorithm of our scheme.

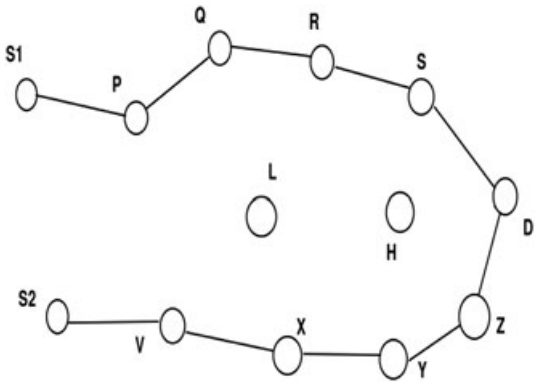

Figure 10. Path under normal flow.

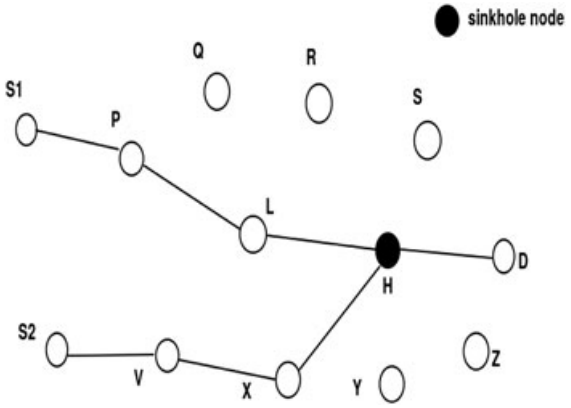

Figure 11. Path under sinkhole attack.

node H. As the node H is malicious sinkhole node, so it can modify, drop, or delay the packets. The newly formed path are $P_1 = \langle S_1, P, L, H, D\rangle$ and $P_2 = \langle S_2, V, X, H, D\rangle$, which are shown in Figure 11. Cluster head (D) computes the overall path $P = P_1 \cup P_2 = \langle S_1, P, L, S_2, V, X, 2H, D\rangle$. Here we can see that H has coefficient value $c_H = 2$ and node H is one hop away ($k_H = 1$) from node D (cluster head). Thus, $k_H < c_H$, and H is considered as sinkhole attacker node.

Remark 2. If the remaining energy under the normal behavior of a node S_i is $REN_{S_{i_\theta}}$ and remaining energy under abnormal behavior of node S_i is REN_{S_i}, the condition $REN_{S_i} < REN_{S_{i_\theta}}$ holds under sinkhole attack.

Example 2. To verify remark 2, consider the scenarios given in Figures 10 and 11. There are two source nodes (S_1 and S_2) which send some information to the destination node D (a cluster head node). The messages are communicated via two paths $P_1 = \langle S_1, P, Q, R, S, D\rangle$ and $P_2 = \langle S_2, V, X, Y, Z, D\rangle$ as shown in Figure 10. Under sinkhole attack, the newly formed paths are $P_1 = \langle S_1, P, L, H, D\rangle$ and $P_2 = \langle S_2, V, X, H, D\rangle$ as shown in Figure 11. So from these observations, we can say that node H receives two requests (packets from source nodes S_1 and S_2). The energy depletion of node H is high as compared with the other intermediate nodes. The threshold value of remaining energy under the normal behavior of node H is REN_{H_θ} and remaining energy under abnormal behavior of node H is REN_H. We can thus say that $REN_H < REN_{H_\theta}$. In general, the relation $REN_{S_i} < REN_{S_{i_\theta}}$ also holds.

3.5.2. Sinkhole node identification algorithm.

Note that a sinkhole node can be of three types, such as SMD, SDP, and SDL. The sinkhole node existence algorithm only gives information about whether the particular node is sinkhole node or a normal node. But to confirm the type of a sinkhole node, we propose the sinkhole node identification algorithm, which is given in Algorithm 2. The SMD are identified by the cluster head CH_j by using HMAC. Suppose for a sinkhole node X a cluster head CH_j receives message MSG_r that is not same as the original message MSG_o which was sent be a source node. Then, there is a mismatch between the hashed HMAC values of MSG_r and MSG_o. In that condition, the cluster head CH_j confirms that sinkhole node X is a SMD node, and CH_j adds it into a list, called SMD_l.

A cluster head CH_j may observe that the messages are delayed by some sinkhole attacker node X as the receiving time of a message T_{MSG_r} is greater than the actual receiving time of the message T_{MSG_o}. In that situations, CH_j checks for the other factors such as network congestion. If it is not there, node X is detected as a *SDL* node, CH_j adds it into another list, called SDL_l.

If a cluster head CH_j does not receive messages from sinkhole node X, it checks whether the node X is a SDP node or there is some other problem in the network such as node failure. CH_j sends the status-cum-data query message M_{sdq} to node X and the waiting time is incremented each time. If waiting time expires and CH_j does not receive any response message (M_{sr}) and data message (M_d) from node X, it is considered that node X fails. Otherwise, if CH_j receives response message (M_{sr}) but it does not receive data message (M_d), node X is detected as the SDP node, and CH_j also adds it into a list, called SDP_l.

These all three conditions are checked by the sinkhole node identification algorithm at the same time. As a result, the algorithm identifies three types of the sinkhole attacker nodes.

4. MATHEMATICAL MODELS FOR THE PROPOSED SCHEME

In this section, we develop the mathematical models for our scheme in terms of the PDR, throughput and EED for the entire network.

4.1. Packet delivery ratio

Let PDR_n, PDR_a, and PDR_s be the PDR under the normal flow, under sinkhole attack, and under our proposed scheme, respectively. Let $|M_d|$ be the number of data packets sent by the cluster members and $|M_{d'}|$ the number of actual packets received by the cluster heads in the network. Further, let $|M_{dsdpa}|$ be the total number of data packets dropped by the SDP in the network, $|M_{dsdpa'}|$ the number of data packets dropped by the SDP nodes true positives (TP) and $|M_{d_1}|$ the number of data packets dropped by the SDP nodes false negative (FN). Further, let $|M_{dsdla}|$ be the total number of data packets delayed by the *SDL* nodes in the network, $|M_{dsdla'}|$ the number of data packets delayed by SDL nodes (TP) and $|M_{d_2}|$ the number of data packets delayed by SDL nodes (FN).

Therefore, it is clear that $|M_{d_1}| = |M_{dsdpa}| - |M_{dsdpa'}|$ and $|M_{d_2}| = |M_{dsdla}| - |M_{dsdla'}|$. Then, under the normal flow, we have,

$$PDR_n = \frac{|M_{d'}|}{|M_d|}.$$

Under the sinkhole attack, PDR can be estimated by

$$PDR_a = \frac{|M_{d'}| - (|M_{dsdpa}| + |M_{dsdla}|)}{|M_d|}.$$

Finally, PDR under our proposed scheme can be also estimated by

$$PDR_s = \frac{|M_{d'}| - (|M_{d_1}| + |M_{d_2}|)}{|M_d|}.$$

4.2. Throughput

Let TH_n, TH_a, and TH_s denote the throughput of the network under the normal flow, under sinkhole attack, and under our proposed scheme, respectively. Suppose T_n, T_a,

DOI: 10.1002/sec

and T_s are the packets delivery time under the normal flow, under sinkhole attack, and under our proposed scheme, respectively. Then, the throughput under normal flow can be estimated as

$$TH_n = \frac{|M_{d'}| \times |pkt|}{T_n}.$$

Similarly, the throughput under attack can be estimated as

$$TH_a = \frac{|pkt| \times (|M_{d'}| - (|M_{dsdpa}| + |M_{dsdla}|))}{T_a},$$

and the throughput under our proposed scheme is given by

$$TH_s = \frac{|pkt| \times (|M_{d'}| - (|M_{d_1}| + |M_{d_2}|))}{T_s},$$

where $|pkt|$ is the size of a data packet.

4.3. End-to-end delay

Let Δ_n, Δ_a, and Δ_s be the EED under the normal flow, under sinkhole attack, and under our proposed scheme, respectively. The EED under the normal flow is given by

$$\Delta_n = \Delta,$$

where Δ can be expressed as

$$\Delta = \frac{\sum_{i=1}^{p}(T_{rec_i} - T_{send_i})}{p},$$

T_{rec_i} and T_{send_i} are the receiving and sending time of a packet i, and p the total number of packets. The EED under the sinkhole attack can be estimated by

$$\Delta_a = \Delta_{n'} + (\Delta_{n_{sdpa}} + \Delta_{n_{sdla}}),$$

where n is the number of sensor nodes in the network, n_{sdpa} the number of the SDP and n_{sdla} the number of the SDL in that cluster, $n' = n - (n_{sdpa} + n_{sdla})$ the number of normal nodes under attack, and $\Delta_{n_{sdpa}}$ the delay corresponding to n_{sdpa} SDP, and $\Delta_{n_{sdla}}$ the delay corresponding to n_{sdla} SDL. Finally, the EED under our proposed scheme is given by

$$\Delta_s = \Delta_{n''} + (\Delta_{FN_{sdpa}} + \Delta_{FN_{sdla}}),$$

where FN_{sdpa} is the number of nodes detected as normal nodes by the proposed scheme, actually they are SDP, FN_{sdla} the number of nodes detected as normal nodes by the proposed scheme, but actually they are SDL, $n'' = n-(n_{FN_{sdpa}}+n_{FN_{sdla}})$ the number of normal nodes under our scheme, $\Delta_{FN_{sdpa}}$ the delay corresponding to FN_{sdpa} nodes and $\Delta_{FN_{sdla}}$ the delay corresponding to FN_{sdla} nodes. Note that if the number of FN nodes is zero, the EED becomes $\Delta_s = \Delta_n$.

5. ANALYSIS OF THE PROPOSED SCHEME

In this section, we analyze the security of our proposed scheme. We further analyze the communication and computational overheads in a cluster.

5.1. Security analysis

In each cluster, there is a powerful node which acts as a cluster head (*CH*). The cluster head is responsible for detecting the all SMD, SDP, and SDL nodes in the corresponding cluster. If a *CH* is compromised by an attacker, that it will be detected by the BS as described in Section 3.2 [48,49], and in such case, we need the deployment of a new cluster head. Each cluster head has the information, such as the identity (ID_{S_i}) of each member (sensor node S_i) and their remaining battery backup REN_{S_i}.

If a sinkhole attacker node is successfully deployed and it starts tampering with data such as modification, dropping and causing unnecessary delay, on the basis of these malicious acts three types of possible sinkhole attacker nodes are SMD, SDL, and SDP nodes. The proposed detection scheme is able to detect all three types of sinkhole attacker nodes at the same time. It works in two phases. In phase 1, it detects the existence of sinkhole attacker node by using sinkhole node existence algorithm (Algorithm 1), which uses the parameters such as node identification ID_{S_i}, path information from source to destination contains hop count k_{S_i} and coefficient of the suspected node c_{S_i}, and remaining energy REN_{S_i} at the nodes. A node S_i is detected as a sinkhole attacker node if $k_{S_i} \leq c_{S_i}$ and $REN_{S_i} < REN_{S_{i_\theta}}$.

After the completion of phase 1, we obtain a common list of all sinkhole attacker nodes contains SMD, SDP, and SDL nodes, but we are not able to identify which node is of which type. To identify different types of sinkhole attacker nodes, we run sinkhole node identification algorithm (Algorithm 2) in phase 2 of the detection process. The SMD nodes are identified by the cluster head CH_j by using HMAC. Suppose for a sinkhole node S_i a cluster head CH_j receives message MSG_r that is not same as the original message MSG_o which was sent by the source node. In that case, there is a mismatch between the hashed HMAC values of MSG_r and MSG_o. In that condition, cluster head CH_j confirms that sinkhole node S_i is a *SMD* node.

If a cluster head CH_j observes that the messages are delayed by some sinkhole attacker node S_i as the receiving time of a message T_{MSG_r} is greater than the receiving time of the original message T_{MSG_o}, CH_j checks the other factor such as network congestion. If there is no congestion, node S_i is detected as a SDL node. If a cluster head CH_j does not receive messages from sinkhole node S_i, it checks whether node S_i, is a SDP node or there is some other problems in the network such as node failure. CH_j sends the status-cum-data query messages M_{sdq} to node S_i, and the waiting time is incremented each time. If waiting time expires and CH_j does not receive response message (M_{sr})

Algorithm 2 Sinkhole node identification algorithm

for each cluster C_j in HWSN **do**
  if for sinkhole node X, HMAC of $MSG_r \neq$ HMAC of MSG_o **then**
    Sinkhole node X modifies the messages and it is detected as an *SMD* node by the cluster head CH_j. CH_j adds node X to the list of sinkhole message modification nodes, SMD_l.
  else if $T_{MSG_r} > T_{MSG_o}$ **then**
    CH_j checks for network congestion.
    if $T_{MSG_r} = T_{MSG_o} + T_{con}$ **then**
      Congestion is detected.
    else
      Sinkhole node X delays the messages and it is detected as an *SDL* node by the cluster head CH_j. CH_j adds node X to the list of sinkhole message delay nodes, SDL_l.
    end if
  else
    Cluster head CH_j does not receive any message.
    Cluster head CH_j sends M_{sdq} to node X.
    Set $WT = WT + 1$.
    if $WT > WT_\theta$ **then**
      if CH_j receives M_{sr} from node X but does not receive any M_d **then**
        Node X is detected as message dropping node by the cluster head CH_j. CH_j adds node X to the list of SDP, SDP_l.
      else
        Cluster head CH_j does not receive M_{sr} and M_d from node X.
        Node failure is detected.
      end if
    end if
  end if
  CH_j blacklist node X, which is detected as *SMD*/*SDL*/*SDP* node, and broadcasts its identity (ID_X) to all its cluster members.
end for

and data message (M_d) from sinkhole node S_i, it is considered that S_i is a failure node. Otherwise, if CH_j receive response message (M_{sr}) but it does not receive data message (M_d), S_i is detected as the SDP node. So we identify the different types of sinkhole attacker nodes in the phase 2 of the detection process. In this way, our scheme has the ability to detect different types of sinkhole attacker nodes in HWSNs.

5.2. Communication cost

To compute the communication cost, we assume that there are n number of sensor nodes in a cluster. Under normal flow, each cluster head CH_j sends n number of status-cum-data query messages M_{sdq} to its cluster members. The members then reply n number of status reply messages to CH_j, and CH_j also gets at most n number of data messages M_d. So total number of exchanged messages under normal flow is $3n$. Under sinkhole attack, CH_j only gets $n - n_{sdp}$ data messages M_d as sinkhole dropping nodes do not send any data messages. The total number of messages exchanged under sinkhole attack is $(n + n + n - n_{sdp}) = 3n - n_{sdp}$. Under the proposed scheme when CH_j does not receive data messages from SDP nodes, it again sends n_{sdp} number of status-cum-data query messages M_{sdq} only to the SDP nodes. SDP nodes send only the status reply messages M_{sr}, but not the data messages M_d. So CH_j only receives n_{sdp} number of M_{sr} messages. After performing both phases of detection, CH_j identifies the sinkhole attacker nodes and also their types. CH_j sends n–(n_{smd}+n_{sdp}+n_{sdl}) information messages to alert the cluster members. CH_j does not send information messages to the SDM, SDP, and SDL nodes. Hence, the total number of exchanged messages under the proposed scheme becomes $[n + n + (n - n_{sdp}) + n_{sdp} + n_{sdp} + n - (n_{sdm} + n_{sdp} + n_{sdl})] = 4n - (n_{sdm} + n_{sdl})$.

5.3. Computation cost

The proposed detection scheme is divided into two phases. In the first phase, we detect the presence of sinkhole attacker nodes in the network by using sinkhole node existence algorithm. If the presence of sinkhole nodes is confirmed, the sinkhole node identification is performed by using sinkhole node identification algorithm in second phase. So the sinkhole node existence algorithm runs first and then the sinkhole node identification algorithm. In the first phase, each CH_j uses the mechanism of ad hoc on demand distance vector (AODV) routing protocol to find out the different paths, which takes $O(2D)$ time [53], where D is the diameter of the network in a cluster. D can be further written in terms of n number of sensor nodes available in a cluster and m number of edges. This can be further written as $O(2nm)$, where $m = n(n-1)/2$ at most. In the real world scenario, this expression can be optimized as $O(n^2)$ [54]. The rest of the steps of the sinkhole node existence algorithm runs in linear time. So the complexity of the sinkhole node existence algorithm is $O(n^2)$. All the steps of the sinkhole node identification algorithm are performed in $O(n)$ time. As a result, the overall time complexity of the proposed scheme is $O(n^2)$ required for a cluster head CH_j.

Remark 3. In our scheme, a sensor node S_i in a cluster needs to send one status response message M_{sr} and another data message M_d, if required, to its corresponding cluster head CH_j. The total number of messages sent by S_i becomes two. Furthermore, S_i requires two HMAC computations for sending the messages M_{sr} and M_d. In addition, to verify status-cum-data query message M_{sdq} S_i needs one more HMAC computation. Because hash function computation is very efficient, the computational cost of S_i is very low. Because of minimum number of message transmissions and low computational cost, our scheme is very effective for the extremely resource-constrained sensor nodes.

6. SIMULATION

This section shows the practical perspective of our scheme using the widely accepted NS2 2.35 simulator [55]. NS2 is a software used for discrete event simulation of a network. It is frequently used in networking research. NS2 provides substantial support for simulation of TCP/UDP protocols, routing protocols (e.g., AODV, DSR etc.) and multicast protocols over wired and wireless networks. NS2 is a standard experiment environment in research community [56].

6.1. Simulation environment

We have simulated our scheme for HWSN on the Ubuntu 14.04 LTS platform using the NS2 simulator. The deployment area is taken as $650 \times 250\ m^2$. In this area, we have deployed 200 nodes such that each cluster consists of a cluster head and 19 sensor nodes. Table III shows the various simulation parameters used in our simulation.

6.2. Simulation scenarios

In network simulation, we simulate the HWSN under normal flow, under sinkhole attack, and under the proposed detection scheme. The scenarios are discussed in the following subsections.

6.2.1. Network scenario under normal flow.

The network scenario under the normal flow contains 200 nodes that are divided into 10 clusters. Each cluster has 20 nodes (19 sensor nodes and one cluster head).

Table III. Simulation parameters.

Parameter	Description
Platform	Ubuntu 14.04 LTS
Deployment area	$650 \times 250\ m^2$
Network topology	Tree
Network size	200 nodes
Number of clusters	10
Number of cluster heads	10
Number of sensor nodes in each cluster	19
Number of attacker nodes	40
Simulation time	1800 seconds
Traffic type	CBR/UDP
Packet size	512 bytes
Packet transmission rate	25 Kbps
Routing protocol	AODV
Medium access control type	IEEE 802.11
Clustering method	Static clustering
Communication range of sensor node	25 m
Communication range of cluster head	50 m

6.2.2. Network scenario under sinkhole attack.

The network scenario under the sinkhole attack again contains 200 nodes that are divided into 10 clusters. Each cluster has 20 nodes (19 sensor nodes and one cluster head). We consider 40 sinkhole attacker nodes. Thus, we have taken 20% nodes in the network as the sinkhole attacker nodes.

6.3. Results and discussion

In our simulation, we have computed the following statistics of the network: (i) packet delivery ratio (PDR), (ii) end-to-end delay (EED) (in ms), (iii) throughput (in kbps), (iv) detection rate (DR), and (v) false positive rate (FPR).

6.3.1. Impact on packet delivery ratio.

Packet delivery ratio is the ratio of number of packet received at the BS to the number of packets sent by the source nodes. Figure 12 shows the PDR under the normal flow, the sinkhole attack and our proposed scheme. From this figure, it is evident that PDR under the normal flow, the sinkhole attack, and our proposed scheme are 0.99, 0.50, and 0.94, respectively. It is then clear that PDR is significantly improved in our scheme as compared with that under the sinkhole attack.

6.3.2. Impact on end-to-end delay.

The EED is the average time taken by the data packets to arrive at the BS. Figure 13 shows the EED (in ms) under the normal flow, the sinkhole attack and our proposed scheme. (EED) is 68.85 ms under the normal flow and 704.86 ms under the sinkhole attack, whereas it is 195.34 ms under our proposed scheme. Thus, (EED) decreases in our scheme.

6.3.3. Impact on throughput.

Throughput is the number of bits transferred per unit time. Figure 14 shows the network throughput (in kbps) under the normal flow, the sinkhole attack as well as our proposed scheme. The throughput is 9.05 kbps under the normal flow, whereas it is 4.52 kbps under the sinkhole

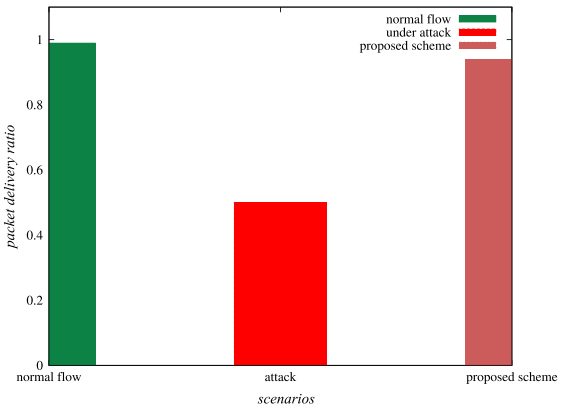

Figure 12. Simulation results of packet delivery ratio

attack and 8.53 kbps under the proposed scheme. It is thus clear that the network throughput is improved significantly (94.25%) in case of our scheme as compared with the sinkhole attack.

Finally, the various statistics of the network in our scheme are provided in Table IV.

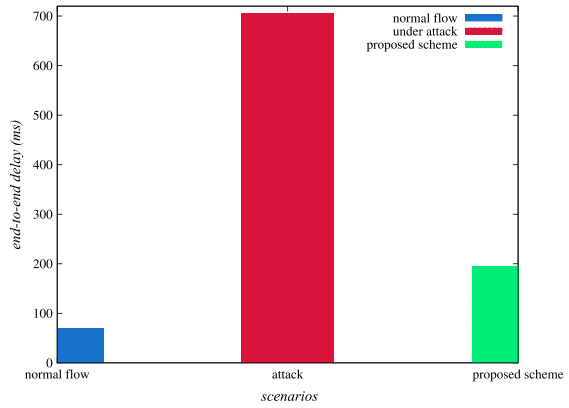

Figure 13. Simulation results of end-to-end delay.

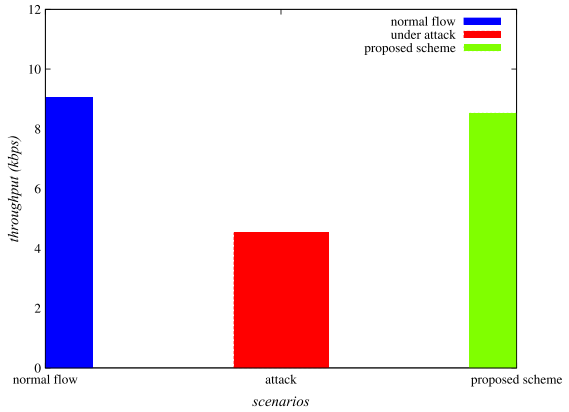

Figure 14. Simulation results of network throughput.

6.3.4. Impact on detection rate and false positive rate.

Two important parameters for our scheme: DR (also called the true positive rate (TPR)) and false positive rate (FPR), are measured in the simulation. These parameters give how an intrusion detection scheme is efficient. Suppose TP denotes the number of true positives, FN the number of false negatives, FP the number of false positives and TN the number of true negatives. Then, DR is measured as the number of attackers detected by the scheme divided by the total number of attackers present in the test set [57], [27], which is given below:

$$DR = \frac{TP}{TP + FN}.$$

FPR is also measured by the number of nodes falsely detected as attacker nodes [57], [27]. *FPR* is defined as

$$FPR = \frac{FP}{TN + FP}.$$

The following observations are then recorded throughout the simulation:

- From the confusion matrix shown in Table V our proposed scheme detects 12 SMD nodes, 10 SDP nodes and 16 SDL nodes. So there are total 38 true positive (TP) nodes (actual attackers), two false positive (FP) nodes (normal nodes), 158 true negative (TN) nodes (normal nodes), and two FN nodes (actually an attacker but detected as a normal node).
- Note that in our network, there are 12 SMD nodes, 11 SDP nodes and 17 SDL nodes and in total 40 sinkhole attacker nodes, and 160 normal nodes. Hence, the DR is 95.0% and the false positive rate (FPR) is 1.25%.

7. PERFORMANCE COMPARISON WITH EXISTING RELATED SCHEMES

In this section, we compare the performance of our scheme with other related existing schemes, such as Salehi *et al.*'s

Table IV. Summary of statistics of the network.

Parameter	Under normal flow	Under attack	Under proposed scheme
Throughput (kbps)	9.05	4.52	8.53
End-to-end delay (ms)	68.85	704.86	195.34
Packet delivery ratio	0.99	0.50	0.94

Table V. Confusion matrix.

		Actual value	
Predicted value	No. of positives	No. of positives TP: 38	No. of negatives FP: 02
	No. of negatives	FN: 02	TN: 158

Table VI. Accuracy comparison.

Parameter	[44]	[27]	[7]	[33]	[9]	[30]	[4]	[5]	Ours
Detection rate (DR)	93.00	90.96	N/A	89.00	N/A	90.00	86.00	83.00	95.00
False positive rate (FPR)	10.00	2.06	1.70	20.00	1.55	N/A	N/A	N/A	1.25

Note: N/A: not available

Table VII. Functionality comparison.

Detection	[16]	[44]	[27]	[7]	[33]	[9]	[30]	[4]	[5]	[45]	Ours
Detection of SMD nodes	Yes	No	No	No	No	No	No	No	No	Yes	Yes
Detection of SDP nodes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
Detection of SDL nodes	No	Yes	Yes	Yes	No	Yes	No	No	No	No	Yes

Note: SMD: sinkhole message modification node, SDP: sinkhole message dropping node and SDL: sinkhole message delay node.

scheme [44], Wang *et al.*'s scheme [27], Hamedheidari *et al.*'s scheme [7], Krontiris *et al.*'s scheme [33], Shafieil *et al.*'s scheme [9], Zhang *et al.*'s scheme [30], Wang *et al.*'s scheme [4] and Wang *et al.*'s scheme [5]. From the results provided in Table VI, it is evident that our scheme has significantly better performance as compared with that for other existing schemes.

We have also provided the functionality comparison in Table VII. From the results provided in this table, it is evident that our scheme has additional detection functionality, and can detect all three types of attacker nodes.

8. CONCLUSION

The different types of sinkhole attacker nodes cause serious threat to the performance of WSN. The existing techniques are not efficient and have limitations as we have already discussed in this paper. We have then proposed new detection technique for the detection of different types of sinkhole nodes for hierarchical WSNs. The presence of different types of sinkhole attacker nodes degrade the performance of the network very rapidly. Under the sinkhole attack, the PDR decreases to 0.50 from 0.99, the network throughput decreases to 4.52 kbps from 9.05 kbps, and the end-to end delay increases to 704.86 ms from 68.85 ms. Therefore, it becomes essential to provide a detection scheme for the sinkhole attack. The network performance parameters are improved significantly in the presence of our proposed scheme. The PDR increases to 0.94, throughput increases to 8.53 kbps and end-to-end delay decreases to 195.34 ms under the deployment of our scheme. Our proposed detection algorithm requires less number of message exchanges, which results in low communication cost. Our scheme significantly achieves high detection rate and low false positive rate as compared with those for other related schemes. In our scheme, the resource-constrained sensor nodes also require the minimum communication and computation overheads as compared with the resource-rich cluster heads in HWSNs. Furthermore, our scheme is secure against sinkhole attack. Therefore, our scheme is suitable for the resource-constrained sensor nodes due to low computation and communication overheads as compared with those for other related schemes.

Acknowledgements

The authors extend their sincere appreciations to the Deanship of Scientific Research at King Saud University for its funding this Prolific Research Group (PRG-1436-16). This work was also supported by the Information Security Education & Awareness (ISEA) Phase II Project, Department of Electronics and Information Technology (DeitY), India. The authors would like to acknowledge the many helpful suggestions of the anonymous reviewers and the Editor, which have improved the content and the presentation of this paper.

REFERENCES

1. Das AK, Sharma P, Chatterjee S, Sing JK. A dynamic password-based user authentication scheme for hierarchical wireless sensor networks. *Journal of Network and Computer Applications* 2012; **35**(5): 1646–1656.
2. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: a Survey. *Computer Networks* 2002; **38**(4): 393–422.
3. Cheng Y, Agrawal DP. An improved key distribution mechanism for large-scale hierarchical wireless sensor networks. *Ad Hoc Networks* 2007; **5**(1): 35–48.
4. Wang Y, Fu W, Agrawal DP. Gaussian versus Uniform Distribution for Intrusion Detection in Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems* 2013; **24**(2): 342–355.
5. Wang Y, Wang X, Xie B, Wang D, Agrawal DP. Intrusion detection in homogeneous and heterogeneous wireless sensor networks. *IEEE Transactions on Mobile Computing* 2008; **7**(6): 698–711.
6. Dong D, Li M, Liu Y, Li X, Liao X. Topological detection on wormholes in wireless ad hoc and sensor

networks. *IEEE/ACM Transactions on Networking* 2011; **19**(6): 1787–1796.
7. Hamedheidari S, Rafeh R. A novel agent-based approach to detect sinkhole attacks in wireless sensor networks. *Computers & Security* 2013; **37**(0): 1–14.
8. Ngai ECH, Liu J, Lyu MR. An efficient intruder detection algorithm against sinkhole attacks in wireless sensor networks. *Computer Communications* 2007; **30** (11-12): 2353–2364.
9. Shafiei H, Khonsari A, Derakhshi H, Mousavi P. Detection and mitigation of sinkhole attacks in wireless sensor networks. *Journal of Computer and System Sciences* 2014; **80**(3): 644–653.
10. Wazid M, Katal A, Sachan RS, Goudar RH, Singh DP. Detection and prevention mechanism for Blackhole attack in Wireless Sensor Network. *International Conference on Communications and Signal Processing (ICCSP 2013)*, Coimbatore, India, 2013; 576–581.
11. Dong D, Li M, Liu Y, Li X, Liao X. Topological detection on wormholes in wireless ad hoc and sensor networks. *IEEE/ACM Transactions on Networking* 2011; **19**(6): 1787–1796.
12. Karlof C, Wagner D. Secure routing in wireless sensor networks: attacks and countermeasures. *Ad Hoc Networks* 2003; **1**(2): 293–315.
13. Tripathi M, Gaur M, Laxmi V. Comparing the Impact of Black Hole and Gray Hole Attack on LEACH in WSN. *International Symposium on Intelligent Systems Techniques for Ad Hoc and Wireless Sensor Networks (IST-AWSN)*, Halifax, Canada, 2013; 1101–1107.
14. Cai J, Yi P, Chen J, Wang Z, Liu N. An Adaptive Approach to Detecting Black and Gray Hole Attacks in Ad Hoc Network. *24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010)*, Perth, Australia, 2010; 775–780.
15. Fessant FL, Papadimitriou A, Viana AC, Sengul C, Palomar E. A sinkhole resilient protocol for wireless sensor networks: performance and security analysis. *Computer Communications* 2012; **35**(2): 234–248.
16. Papadimitriou A, Fessant FL, Viana AC, Sengul C. Cryptographic protocols to fight sinkhole attacks on tree-based routing in Wireless Sensor Networks. *5th Workshop on Secure Network Protocols (NPSec 2009)*, USA, Princeton, 2009; 43–48.
17. Chatterjee S, Das AK. An effective ECC-based user access control scheme with attribute-based encryption for wireless sensor networks. *Security and Communication Networks* 2015; **8**(9): 1752–1771.
18. Das AK. A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks. *Peer-to-Peer Networking and Applications* 2016; **9**(1): 223–244.
19. Das AK. A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor. *International Journal of Communication Systems* 2015, DOI: 10.1002/dac.2933.
20. Das AK. A secure and efficient user anonymity-preserving three-factor authentication protocol for large-scale distributed wireless sensor networks. *Wireless Personal Communications* 2015; **82**: 1377–1404.
21. Shen J, Tan H, Wang J, Wang J, Lee S. A novel routing protocol providing good transmission reliability in underwater sensor networks. *Journal of Internet Technology* 2015; **16**(1): 171–178.
22. Xie S, Wang Y. Construction of tree network with limited delivery latency in homogeneous wireless sensor networks. *Wireless Personal Communications* 2014; **78**(1): 231–246.
23. He D, Zeadally S, Kumar N, Lee JH. Anonymous authentication for wireless body area networks with provable security. *IEEE Systems Journal* 2016, DOI: 10.1109/JSYST.2016.2544805.
24. He D, Zeadally S, Wu L. Certificateless public auditing scheme for cloud-assisted wireless body area networks. *IEEE Systems Journal* 2015, DOI: 10.1109/JSYST.2015.2428620.
25. Dolev D, Yao AC. On the security of public key protocols. *IEEE Transactions on Information Theory* 1983; **29**(2): 198–208.
26. Das ML. Two-factor user authentication in wireless sensor networks. *IEEE Transactions on Wireless Communications* 2009; **8**(3): 1086–1090.
27. Wang SS, Yan KQ, Wang SC, Liu CW. An integrated intrusion detection system for cluster-based wireless sensor networks. *Expert Systems with Applications* 2011; **38**(12): 15234–15243.
28. Zhu WT, Zhou J, Deng RH, Bao F. Detecting node replication attacks in wireless sensor networks: a survey. *Journal of Network and Computer Applications* 2012; **35**(3): 1022–1034.
29. Rajasegarar S, Leckie C, Palaniswami M. Hyperspherical cluster based distributed anomaly detection in wireless sensor networks. *Journal of Parallel and Distributed Computing* 2014; **74**(1): 1833–1847.
30. Zhang FJ, Zhai LD, Yang JC, Cui X. Sinkhole attack detection based on redundancy mechanism in wireless sensor networks. *Procedia Computer Science* 2014; **31**: 711–720.
31. Sreelaja NK, Pai GAV. Swarm intelligence based approach for sinkhole attack detection in wireless sensor networks. *Applied Soft Computing* 2014; **19**: 68–79.
32. Nahas HA, Deogun JS, Manley ED. Proactive mitigation of impact of wormholes and sinkholes on routing

security in energy-efficient wireless sensor networks. *Wireless Networks* 2009; **15**(4): 431–441.
33. Krontiris I, Dimitriou T, Giannetsos T, Mpasoukos M. Intrusion Detection of Sinkhole Attacks in Wireless Sensor Networks. *Third International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2007), Lecture Notes in Computer Science*, Springer, Wroclaw, Poland, 2008; 150–161.
34. Garofalo A, Sarno CD, Formicola V. Enhancing Intrusion Detection in Wireless Sensor Networks through Decision Trees. *Proceedings of 14th European Workshop on Dependable Computing (EWDC 2013), Lecture Notes in Computer Science*, Springer, Coimbra, Portugal, 2013; 1–15.
35. Giruka VC, Singhal M, Royalty J, Varanasi S. Security in wireless sensor networks. *Wireless Communications and Mobile Computing* 2008; **8**(1): 1–24.
36. Hai TH, Huh EN, Jo M. A lightweight intrusion detection framework for wireless sensor networks. *Wireless Communications and Mobile Computing* 2010; **10**(4): 559–572.
37. Du X, Guizani M, Xiao Y, Chen HH. Two tier secure routing protocol for heterogeneous sensor networks. *IEEE Transactions on Wireless Communications* 2007; **6**(9): 3395–3401.
38. Dallas D, Leckie C, Ramamohanarao K. Hop-Count Monitoring: Detecting Sinkhole Attacks in Wireless Sensor Networks. *15th IEEE International Conference on Networks (ICON 2007)*, Adelaide, Australia, 2007; 176–181.
39. Roy SD, Singh SA, Choudhury S, Debnath NC. Countering sinkhole and black hole attacks on sensor networks using Dynamic Trust Management. *Symposium on Computers and Communications (ISCC 2008)*, Morocco, 2008; 537–542.
40. Krontiris I, Giannetsos T, Dimitriou T. Launching a Sinkhole Attack in Wireless Sensor Networks; The Intruder Side. *International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2008)*, Avignon, France, 2008; 526–531.
41. Chen C, Song M, Hsieh G. Intrusion detection of sinkhole attacks in large-scale wireless sensor networks. *International Conference on Wireless Communications, Networking and Information Security (WCNIS 2010)*, Beijing, China, 2010; 711–716.
42. Zhan G, Shi W, Deng J. Design and implementation of TARF: A Trust-Aware Routing Framework for WSNs. *IEEE Transactions on Dependable and Secure Computing* 2012; **9**(2): 184–197.
43. Qi J, Hong T, Xiaohui K, Qiang L. Detection and defence of Sinkhole attack in Wireless Sensor Network. *14th IEEE International Conference on Communication Technology (ICCT 2012)*, Chengdu, China, 2012; 809–813.
44. Salehi SA, Razzaque MA, Naraei P, Farrokhtala A. Detection of sinkhole attack in wireless sensor networks. *International Conference on Space Science and Communication (IconSpace 2013)*, Melaka, Malaysia, 2013; 361–365.
45. Sharmila S, Umamaheswari G. Detection of Sinkhole Attack in Wireless Sensor Networks Using Message Digest Algorithms. *International Conference on Process Automation, Control and Computing (PACC 2011)*, Coimbatore, India, 2011; 1–6.
46. Das AK. An unconditionally secure key management scheme for large-scale heterogeneous wireless sensor networks. *First International on Communication Systems and Networks and Workshops (COMSNETS 2009)*, IEEE, Bangalore, India, 2009; 1–10.
47. Das AK. An efficient random key distribution scheme for large-scale distributed sensor networks. *Security and Communication Networks* 2011; **4**(2): 162–180.
48. Das AK. A random key establishment scheme for multi-phase deployment in large-scale distributed sensor networks. *International Journal of Information Security* 2012; **11**(3): 189–211.
49. Zhu B, Setia S, Jajodia S, Roy S, Wang L. Localized multicast: Efficient and distributed replica detection in large-scale sensor networks. *Mobile Computing, IEEE Transactions on* 2010; **9**(7): 913–926.
50. Wang J, Dong W, Cao Z, Liu Y. On the delay performance in a large-scale wireless sensor network: Measurement, analysis, and implications. *IEEE/ACM Transactions on Networking* 2015; **23**(1): 186–197.
51. Ghazvini M, Vahabi M, Rasid M, Abdullah R, Musa W. Low Energy Consumption MAC Protocol for Wireless Sensor Networks. *2nd IEEE International Conference on Sensor Technologies and Applications*, Cap Esterel, France, 2008; 49–54.
52. Park S, Hong S W, Lee E, Kim SH, Crespi N. Large-scale mobile phenomena monitoring with energy-efficiency in wireless sensor networks. *Computer Networks* 2015; **81**: 116–135.
53. Lee A, Ra I, Kim H. Performance Study of Ad Hoc Routing Protocols with Gossip-based Approach. *Proceedings of the 2009 Spring Simulation Multiconference (SpringSim '09)*, San Diego, USA, 2009; 1–8.
54. Crescenzi P, Grossi R, Habib M, Lanzi L, Marino A. On computing the diameter of real-world undirected graphs. *Theoretical Computer Science* 2013; **514**: 84–95.
55. The Network Simulator- ns-2. Available at http://www.isi.edu/nsnam/ns/. [Accessed on March 2015.]

56. Wang J. NS-2 Tutorial. Available at http://www.cs.virginia.edu/~cs757/slidespdf/cs757-ns2-tutorial1.pdf. [Accessed on March 2015.]

57. Kasliwal B, Bhatia S, Saini S, Thaseen IS, Kumar CA. A hybrid anomaly detection model using G-LDA. *International Advance Computing Conference (IACC 2014)*, Gurgaon, India, 2014; 288–293.