I. J. Computer Network and Information Security, **2015, 3, 50-56**
Published Online February 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2015.03.07

Detecting Sinkhole Attacks in Wireless Sensor Network using Hop Count

Md. Ibrahim Abdullah
Computer Science and Engineering, Islamic University, Kushtia, Bangladesh
Email: ibrahim25si@yahoo.com

Mohammad Muntasir Rahman and Mukul Chandra Roy
Computer Science and Engineering, Islamic University, Kushtia, Bangladesh
Email: {shohan6, ry_mkl}@yahoo.com

Abstract—Nowadays, Wireless Sensor Networks (WSNs) are widely used in many areas, especially in environment applications, military applications, queue tracking, etc. WSNs are vulnerable to different types of security attacks due to various constraints such as broadcasted nature of transmission medium, deployment in open or hostile environment where they are not physically protected, less memory, and limited battery power. So, security system is the crucial requirements of these networks. One of the most notably routing attacks is the sinkhole attack where an adversary captures or insert nodes in the sensor field that advertise high quality routes to the base station. In this paper, a mechanism is proposed against sinkhole attacks which detect malicious nodes using hop counting. The main advantage of the proposed technique is that, a node can detects malicious nodes only collaborating with the neighbor nodes without requiring any negotiation with the base station. Simulation result shows that, the proposed technique successfully detects the sinkhole nodes for large sensor field.

Index Terms—Sinkhole Attack, Wireless Sensor Network, Routing Attack, Hop distance.

I. Introduction

Wireless sensor networks (WSNs) are an emerging technology consisting of small, low-power devices that integrate limited computation, sensing and radio communication capabilities. The main objectives of deploying the Wireless Sensor Network are remote monitoring and gathering information [1]. WSNs are typically used out in an open, uncontrolled environment, often in hostile territories. However, the open nature of the wireless communication channels, the lack of infrastructure, the fast deployment practices, and the hostile environments where sensor nodes are deployed, make them vulnerable to a wide range of security attacks [2] [3]. In particular, several important applications for such networks come from military and defense arenas. For example, in emergency response operations such as after a natural disaster like a flood, tornado, or earthquake, a wireless sensor network could be used for real time feedback. Therefore, the emergency rescue will rely on that particular type of network.

Use of wireless medium and inherent collaborative nature of the network protocols make such network vulnerable to various forms of attacks [3]. Several types of malicious attacks have been well described in the literatures [2][3][4]. Attackers can eavesdrop on radio transmissions, inject bits in the channel and replay previously heard packets. The adversary may deploy few malicious nodes with similar hardware capabilities as the legitimate nodes. The attackers may come upon these malicious nodes by purchasing them separately or by capturing legitimate nodes and physically overwriting their memory. Moreover, defense techniques used in wire networks are hard to apply in wireless sensor network with limited processing power and resources. An adversary may disable a WSN by interfering with intra-network packet transmission via sinkhole attacks [2], Sybil attacks [2], jamming or packet injection attacks [3], wormhole attacks [5]. This work focuses on sinkhole attacks [2][6][7].

In a sinkhole attack, the goal of an adversary is to lure nearly all the traffic from a particular area through a captured node, creating a metaphorical sinkhole with the adversary at the center. Because nodes on, or near, the path that packets follow have many opportunities to tamper application data, sinkhole attacks can enable many other attacks (selective forwarding, for example). Sinkhole attacks typically work by making a compromised node look especially attractive to surrounding nodes with respect to the routing algorithm. For instance, an adversary could spoof or replay an advertisement for an extremely high quality route to a base station [2]. The transmission of this routing advertisement lets each neighboring node of the attacker forward the packets intended to the base station through this attacker. For example, a laptop-class adversary has a strong power radio transmitter that allows it to provide a high-quality route by transmitting with enough power to reach base station (BS) in a single hop, or by using a wormhole attack. WSNs are particularly susceptible to sinkhole attack is due to their specialized communication pattern. Since all packets share the same ultimate

destination (many-to-one communication model for single BS), a compromised node needs only to provide a single high quality route to the BS.

This paper introduces an easy and effective method to detect and locate sinkhole nodes. In proposed technique when a sinkhole node broadcast shortest hop distance from base station, the neighbors of this node compare the lowest hop distance with a database of hop distance. The database is created in network initialization phase. If it is remarkably low than conclusions can be made that there may exists sinkhole attacks.

The remainder of this paper is organized as follows. In Section II, we formally describe the sinkhole attack in wireless sensor networks. Section III presents the related work. In Section IV, we present our detection algorithm. The performance of the proposed algorithm is evaluated in Sections V through simulations. Finally, Section VI concludes this paper.

II. Problem Statement

In this work, we consider that sensor nodes are deployed in an open place and do not contain any tamper proof hardware. The nodes may be compromised. An attacker can capture sensor nodes and can extract all key material, data, and code stored on that node, which was previously a legitimate member of the network. She can reprogram the memory of the capture nodes using a laptop that the node has a high-quality single-hop link to the base station (BS). It can then broadcast routing messages about the high quality route, thus spoofing the surrounding nodes to create a sinkhole.

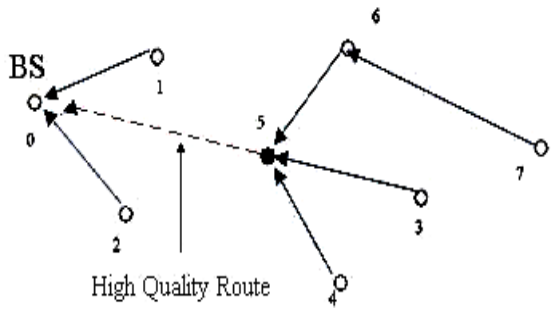

Fig 1. Sinkhole Attack

For example, as shown in fig. 1, adversary node 5 advertises a one hop-count route to the BS. As a result, node 3, 4, 6 and 7 select node 5 to relay their data. The attacker may drop their packets causes Denial of Service (DoS) for four nodes or change the contents of the packets and resend them to BS or other nodes. A sinkhole can also be performed using a wormhole [5], which creates a metaphorical sinkhole. An example is shown in fig. 2, where an adversary creates a sinkhole by tunneling messages received in one part of the network and replays them in a different part using a wormhole. Adversary could convince its surrounding nodes who would normally at multiple hop distance from base station that they are only one or two hops away via the wormhole.

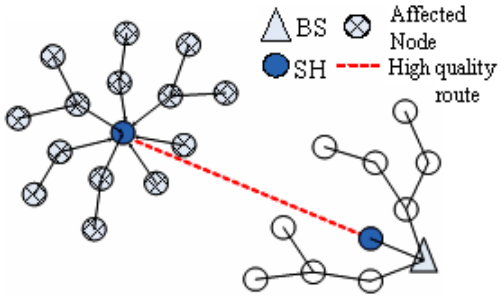

Fig 2. Wormhole Attack [6]

III. Related Work

Intrusion detection has been an active research topic for wireless network especially for wireless ad hoc networks [8]. But sensor networks are different than ad hoc networks. The nodes involved in a WSN are mainly identical in hardware and are designed aiming at an extremely low-cost for a large amount of deployment [1]. These nodes are usually even more resource constrained than most ad-hoc network nodes, with less memory and computation power in order to achieve lower cost and longer battery life. For sensor networks, some existing secure or geographical routing protocols are resistant to sinkhole attack in certain level. An example is a geographic protocol [9], which performs routing by the localized information and interactions only, without an initiation from the base station. However, many of the existing routing protocols, in particular, those based on route advertisement, are vulnerable to sinkhole attacks.

A first approach on the detection of sinkhole attacks in WSN has been presented by Ngai et. al. [6]. This approach involves the base station in the detection process, resulting in a high communication cost for the protocol. The base station floods the network with a request message containing the IDs of the affected nodes. The affected nodes reply to the base station with a message containing their IDs, ID of the next hop and the associated cost. The received information is then used from the base station to construct a network flow graph for identifying the sinkhole. To avoid tampering of packets during transmission, encryption and path redundancy is proposed.

Choi et. al. [7] proposed a detection scheme for sinkhole attacks based on Link Quality Indicator in sensor networks. The proposed method can detect a sinkhole attack that uses LQI based routing and several detecting nodes. General nodes collect minimum link cost between neighborhood node and detecting nodes compute the minimum path cost with surrounding detector nodes in the proposed method. It can detect an abnormally strong signal from the actions of the malicious node by referring to the minimum link cost table. Other existing protocols build detecting mechanisms for sinkhole attacks in sensor networks that are based on routing protocols usually deployed in Ad hoc networks, like the AODV [10] and the DSR Protocol [11].

IV. Proposed Method

This work considers the following network model.

A. Network Model

In this work, we consider a sensor network that consists of a single BS. The network nodes are randomly deployed within a specific region. The node position is static that means it does not change after deployment and all nodes are uniquely identified. The sensor nodes continuously collect and send data to the base station by forwarding packets hop-by-hop. The nodes do not contain any tamper proof hardware, so it may be compromised.

We assume that the BS is located outside from the sensor field in a safe place for processing the sensors reading to draw conclusions. Base station keeps record of all nodes ID. If any node replaced or deployed the record is updated. We also assume that an adversary launch sinkhole attacks by compromising legitimate node/nodes that providing a high quality route to the base station. Only the base station maintains a global view of the location of nodes by some localization mechanisms [12]. It broadcast authenticated beacons to all the nodes in the network periodically. This prevents nodes from recognizing the base stations wrongly.

B. Proposed Detection Technique

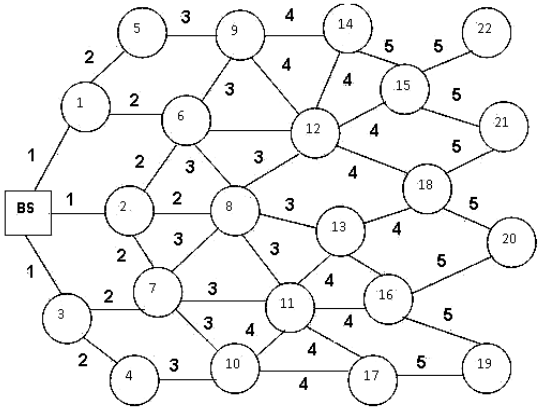

Fig 3. Network without any attack

In proposed detection technique, at first the base station sends a HELLO packet to its nodes. This packet sends to it nearest node say node 1, 2 and 3 as shown in fig. 3. These nodes disseminate packet through the network. The HELLO packet contains a field hop-count. Hop-count specifies the hop distance of the node from BS. The hop-count value for BS is 0. When a node has received packet of hop-count value 0, it deduce that the BS sends the packet. It increases the hop-count value by 1. For example, when Node-2 (Fig. 3) broadcast this packet to other nodes the hop-count value is 1. Node-2 is one hop distance from BS. As the packets disseminate through the network the hop-count value gradually increases. All nodes keep this value in a node neighbor database. The database has at least two entries – node ID and hop-count. After receiving all packets from neighbors, a node sorts the hop-count values. It then calculates the average hop-count value without considering the lowest hop-count and compares average and lowest. If this lowest value is abnormally small comparing with average hop-count, certainly this is anomaly. Because we assume that a node and its neighbors have comparable distance from base station and have alike hop-count.

The main steps involved in the sinkhole detection technique are described below.

Phase 1: Neighbor Database Construction

- **Base Station :**The BS sends a HELLO packet to its nearest nodes. Initially, the hop count of the base station is zero. The message contains the ***Node_ID*** of the sender, and the ***Hop_Count*** (The hop count is the minimum number of node–to–node transmissions to reach a data packet from the node to the base station). The message frame format is shown in the fig. 4.

Node_ID	Hop_Count

Fig 4. Message frame format

- **Sensor Nodes:** When a node receive message from the BS; it assign its ID and hop-count from BS to the message's ***Node_ID*** and ***Hop_Coun***t field respectively. Nodes received such packet directly from base station put *1* in hop-count field and then re-send the message to its nearest neighbor nodes. Nodes that receive such message, keeps the sending node ID and hop-count in its neighbor database. It then sends to its neighbors within radio range '*r'* by increasing the message hop-count value. A node keeps all such messages that it has received from neighbor nodes. This process is continuing to end line of the sensor field. The entries of the database are neighbor node ID and the hop distance from BS. For example, Table 1 shows the neighbor database of node 15 in accordance of fig. 2. Though node 22 is neighbor of node 15 but it only receive message through node 15, for this reason, there is not any entry for node 22 in the database.

Table 1. Neighbor Database of node 15 of Fig. 2

Node_ID	*Hop_Count*
14	5
21	5
12	4
18	5

Phase 2: Sinkhole Node Detection

To clarify the methodology, we examine the effect sinkhole on the network. In a sinkhole attack, the adversary node claim the comparatively shortest root than others nodes around its neighbor as shown in fig. 1. Since each node has limited resources and can not store global information, a node can only use local information to detect sinkhole attacks.

To detect sinkhole attack, at first each node sorted its database using any sorting algorithm. After sorting its neighbor database, the format of database of Node-15 is shown in Table 2. Now, it takes apart the lowest hop-count value and corresponding node ID. If there are several copies of lowest hop-count it separates them all. After splitting the lowest values nodes calculates the average hop-count of the rest others. For our example Node-15, the average hop-count is 5 and the lowest hop-count of the separated neighbor is 4.

Table 2. Neighbor database of node 15 after shorting

Node_ID	*Hop_Count*
12	4
14	5
18	5
21	5

To take a decision about a sinkhole node, it calculates the differences (%) between average and lowest hop-count (Eq.-1). If the difference is greater than a threshold value, it is an anomaly. For multiple value of minimum hop count, all nodes that claim this minimum values will identified as suspicious nodes.

$$Difference(\%) = \frac{AverageHop - MinimumHop}{AverageHop} \times 100\% \quad (1)$$

Our sinkhole detection technique is described in the following steps.

Neighbor database (DB_{ni}) creation

1. The BS disseminates a message to its nearest nodes. The hop-count (hc_i) of this message is 0.
2. Each node when receive packet from base station, upgrade the hop-count field by one and sends to its nearest neighbor nodes.
3. Neighbor nodes correct the hop count value and retransmit to its all neighbors except the sending node.
4. The nodes create a database of all such receive messages. The entries of the database are Node ID and correct value of hop-count.
5. The process is continuing to end line of the sensor field.

Pseudo code of neighbor database

```
1.  for node n: i to N
2.     open database(DB_ni);
3.     for node j to N (j≠i)
4.        if (receive Hello_Msg_j)
5.           correct hc_i = hc_j + 1;
6.           insert value of ID_j and hc_i in DB_ni;
7.        end if
8.        insert node ID_i and hc_i in the Hello_Msg_i;
9.        send Hello_Msg_i to all n;
10.    end for
11. end for
```

Detection Technique: After creation of node neighbor database (DB_{ni}) with hop-count

1. Node shorted its DBni on hop count from base station.
2. Separate the lowest value of hop-count (hc_{li}) and node-ID (ID_l).
3. The node makes an average hop-count (hc_{avi}) excluding the lowest value.
4. Compares the average hop distance with lowest hop distance.
5. If it is greater than threshold (T_h), node activity is suspicious.
6. Broadcast this to other nodes and inform BS about the node.

Pseudo code of Detection Algorithm

```
1.  initialize threshold T_h;
2.    for node n: i to N
3.      sort DB_ni by hop-count hc_i;
4.      lowest hop-count hc_li = minHop(DB_ni);
5.      lowest hop-count node ID_l = ID(hc_li);
6.      calculate average hc_avi = AvgHop(DB_ni - hc_li);
7.      calculate dif = [(hc_avi - hc_li)/ hc_li]×100;
8.      if (dif >=T_h)
9.          return "Sinkhole Found" to base station;
10.   end if
11. end for
```

To illustrate our proposed technique we create a sinkhole node Node–14 of fig. 2. Node – 14 is the neighbor of Node–15. Node–14 claims that it is two hop distance from base station (Table 3). Now we examine how the Node–15 can detect the malicious activity of Node–14.

Table 3. Sorted neighbor database of node 15 after changing the hop-count of node 14

Node_ID	*Hop_Count*
14	2
12	4
21	5
18	5

To detecting malicious node or wormhole node, we assumed that the difference threshold is 40%. The average hop count value without considering the minimum hop count for Node–15 is 4.67. The difference in percent is 57%. This value is greater than the difference threshold. So Node–15 is successful to detect the suspicious activity of Node–14. Now Node–15 broadcast this malicious activity to inform base station about Node–14.

We define success rate of our detection technique as how many neighbor nodes of malicious Node–14 can identified this malicious activity. If any neighbor of Node–14 failed to identify the malicious activity then it is unsuccessful detection. So the percentage of success and failure depends on number of neighbors of a sinkhole node.

V. Simulation And Results

In order to verify the success rate of our detection technique, we simulated our proposal. Our simulation is conducted over a 100m×100m rectangular flat space with randomly distributed 100 sensor nodes. Table 4 presents the simulation parameters. The deployed nodes have fixed positions during the entire simulation time. Randomly some nodes are selected as sinkhole nodes and their hop-count are manually changed. We consider two ray ground propagation model of radio wave [13].

Table 4. Simulation parameters

Sensor Field Area	100m × 100 m
Number of Nodes	100
Transmission Power	-5 dBm
Threshold Value	10% - 90%
Radio Range (*r*)	17 m
Node Type	Mica2

We assume that an adversary needs sometime to capture a node, connect it with a laptop/computer system and extract information [14]. She may be able to capture or insert node/nodes when sensor nodes create the neighbor database. To measure the success rate of the proposed technique, we have compared the average hop distance with lowest hop distance. Usually a node at far distance from base station has more hop-count. These nodes are more susceptible to sinkhole attack. For that reason at first we take simulation for different threshold value to find an optimum value of threshold. Then we used this value to find the relationship of successful detection with node distance from base station.

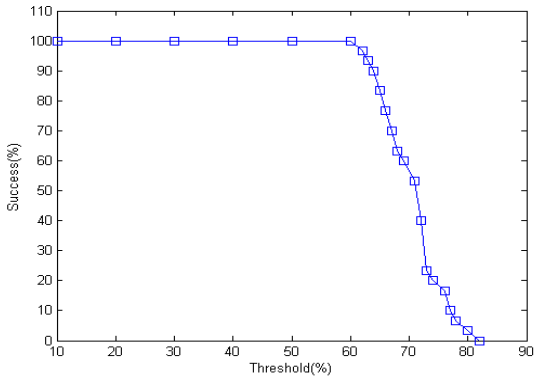

Fig 5. Detection rate based on threshold when difference is 3 hops

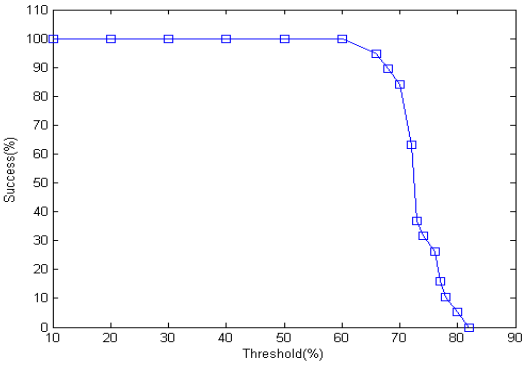

Fig 6. Detection rate based on threshold when difference is 2 hops

The fig. 5 and fig. 6 show sinkhole node detection rate with respect to different threshold values when minimum difference between average and lowest hop-count is 3 and 2 hops, respectively. These two curves are similar in look. In these two cases, the detection rate is 100% within range of the threshold value from 10% to 60%. After then it decreases gradually with increasing thresholds value. It is because we set the difference between average and lowest hop-count. The minimum difference between average and lowest hop-count is one. If we set this minimum difference, all nodes in one hop distance are identified as malicious sinkhole node.

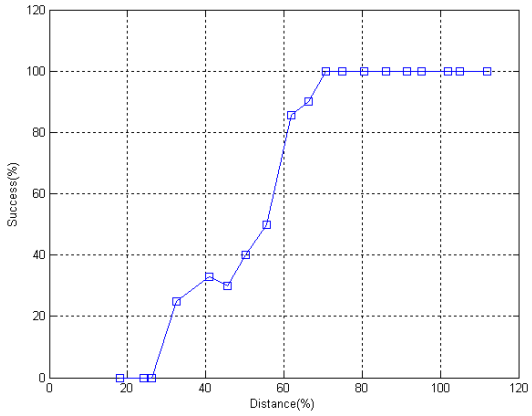

Fig 7. Detection rate based on distance when minimum hop difference is 3

To find the relation of proposed technique with node position at first we take threshold as 50%. From fig. 5 it is found that the detection rate decreases above 60% of threshold. For an optimum we take 50% as threshold. The fig. 7 and fig. 8 show the detection rate with respect to the position of nodes from base station for minimum hop difference 3 and 2 respectively.

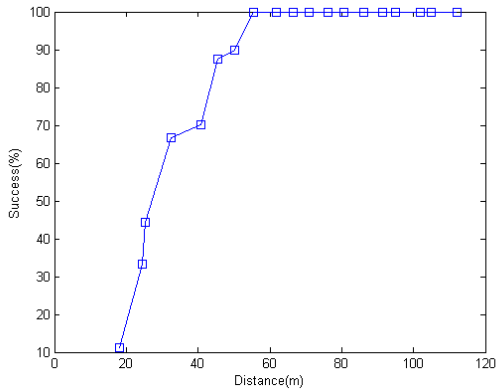

Fig 8. Detection rate based on distance when minimum hop difference is 2

In this case, the detection is 0% near to the base station. Since within 17 meter of radio range is one hop, all nodes have equal hop distance whatever they are malicious or legitimate. As the node distance from base station increases, sinkhole node detection rate increase. Because hop-to-hop communication model nodes at far distance have more hop-count than the nodes near to base station. The detection rate is 100% when a node locates at 70 meter distance from base station for minimum three hop difference (Fig. 5). For two hop difference we found similar curve as in fig. 5 except the detection rate reached at 100% at about 60 meter from base station.

VI. Conclusion

In this paper, we have presented a new algorithm to detect sinkhole attacks in wireless sensor network. Our proposed technique uses a hop counting technique for detecting sinkhole nodes. Proposed technique does not require additional hardwire, node location or send any information to base station. The computation complexities are sorting the hops and average all hop-count. These computations fit well with present architecture of sensor node. There is not any extra communication in proposed technique. The hop distance of a node from base station is common technique for all routing protocols. When the malicious node position is near to base station (one or two hop distance), our algorithm can not accurately detect sinkhole nodes. The detection technique can be increased for lower threshold value but it introduces false detection.

The proposed technique successfully detects the sinkhole attack when this malicious node located at far distance from base station. The technique is also applicable to wormhole attack as the attack is almost similar to sinkhole attack. Proposed technique is also applicable when sinkhole nodes advertise high quality link, strong transmitted power etc. In those cases, we have to sort the advertising parameter and take decision which value is strange.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “A survey of sensor networks”, *IEEE Communications*, vol. 40(8), pp. 102-114, 2002.

[2] C. Karlof and D. Wagner, “Secure routing in wireless sensor networks: Attacks and countermeasures,” in *Proceedings of 1st IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003. pp. 293-301.

[3] A. Wood and J. Stankovic, “Denial of service in sensor networks”, *IEEE Computer*, vol. 35(10), pp. 54-62, 2002.

[4] T. Roosta, S. Shieh and S. Sastry, “Taxonomy of Security Attacks in Sensor Networks and Countermeasures”, Berkeley, California, University Press.

[5] Y. Xu, G. Chen, J. Ford and Fillia Makedon, “Detecting Wormhole Attacks in Wireless Sensor Networks” *IFIP International Federation for Information Processing*, Volume 253, Critical Infrastructure Protection, Pages 267-279, 2007.

[6] E. C. H. Ngai, J. Liu and M. R. Lyu, "On the Intruder Detection for Sinkhole Attack in Wireless Sensor Networks," *in Proc. IEEE ICC., pp. 3383-3389*, June 2006.

[7] B. G. Choi, E. J. Cho, J. Ho Kim, C. S. Hong, and J. H. Kim, "A Sinkhole Attack Detection Mechanism for LQI based Mesh Routing in WSN," *in Proc. ICOIN 2009*, pp. 1-5, Jan 2009.

[8] Y. Zhang and W. Lee, “Intrusion Detection in Wireless Ad-Hoc Networks,” in *Proc. of the 6th ACM MobiCom*, Aug 2000, pp. 275-283.

[9] B. Karp and H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks,” in *Proc. of the 6th ACM MobiCom*, Aug 2000, pp. 243-254.

[10] D. Dallas, C. Leckie, and K. Ramamohanarao, “Hop-count monitoring: Detecting sinkhole attacks in wireless sensor networks,” in *ICON ’07: Proceedings of the 15th IEEE International Conference on Networks*, Adelaide, SA, 2007, pp. 176–181.

[11] A. A. Pirzada and C. McDonald, “Circumventing sinkholes and wormholes in wireless sensor networks,” in *IWWAN ’05: Proceedings of International Workshop on Wireless Ad-hoc Networks*, 2005.

[12] L. Hu and D. Evans, “Localization for Mobile Sensor Networks,” in *Proc. of the 10th ACM MobiCom*, Sep 2004, pp. 45-57.

[13] T. S. Rappaport. “Wireless communications: principles and practice”, Prentice Hall, 2nd edition, 2002.

[14] Fei Hu, Waqaas Siddiqui, Krishna Sankar, “Scalable security in Wireless Sensor and Actuator Networks (WSANs): Integration re-keying with routing”, *Computer Networks, Science Direct, Elsevier,* Vol. 51 (2007), pp 285–308.

Authors’ Profiles

Md. Ibrahim Abdullah has been serving as Associate Professor of the Department of Computer Science and Engineering, Islamic University, Kushtia, Bangladesh. His areas of interest include Network security, Wireless Sensor Network, Cognitive Radio and wireless communication.

Mohammad Muntasir Rahman has been received his B.Sc and M.Sc from the Department of Computer Science & Engineering at Islamic University, Bangladesh and currenty he is an Assistant Professor of the same department at Islamic University, Kushtia, Bangladesh. His areas of interest include Wireless Sensor Network, Cognitive Radio Network.

Mukul Chandra Roy received his B.Sc and M.Sc. from Department of Computer Science and Engineering at Islamic University, Kushtia, Bangladesh. Currently he is an Assistant Hardware Maintenance Engineer at Janata Bank, Bangladesh. His areas of interest include Network security and wireless communication.

How to cite this paper: Ibrahim Abdullah, Mohammad Muntasir Rahman, Mukul Chandra Roy,"Detecting Sinkhole Attacks in Wireless Sensor Network using Hop Count", IJCNIS, vol.7, no.3, pp.50-56, 2015.DOI: 10.5815/ijcnis.2015.03.07

