The Sinkhole Attack: Severity Analysis & Countermeasures in Wireless Sensor Network

Hasibul Hasan Mansoori[1] Mr. Mukesh Kumar[2] Mr.Pritaj Yadav[3]
[1]Assistant Professor [2]Head of Department [3]Associate Professor
[1,2,3]Department of Computer Science & Engineering
[1]Maulana Azad College of Engineering & Technology, Neoraganj, Neora, Patna, India [2,3]Rabindra Nath Tagore University Raisen, M.P, India

Abstract— Wireless sensor networks have been recognized as being valuable in a range of fields to include mainly military sensing and tracking, environment monitoring, patient monitoring and tracking smart environment. Sensor network possesses unique challenges to protocol builders, because these tiny wireless devices are often deployed in unattended environment with limited potentials. Therefore these networks are susceptible to different types of malicious attacks. In this paper, we will discuss on sinkhole attacks, analysis and its countermeasures. In this attack, the adversary's goal is to attract nearly all the traffic from a particular area through a compromised node.

Key words: Wireless Sensor Networks, Sinkhole Attack, Analysis, Countermeasures

I. INTRODUCTION

The wireless sensor network (WSN) consists of a large number of sensor nodes, which are extremely small, low power, and low cost miniature devices built using semiconductor manufacturing techniques. Wireless sensor networks may either monitor or control systems that may consist of thousands of nodes deployed in very high density. They could be located in homes and buildings, highways and cities, in infrastructures. WSN may be used for monitoring and warning of natural disasters and the affects including floods, winds and other natural phenomenon. They could be used for conducting military surveillance. Wireless sensor nodes are also called motes. Sensor nodes have capability to collect sensed data and send that to the base station, a WSN generally consist of a base station that can communicate with a number of wireless sensors via radio link. WSN uses a wireless channel to communicate, so there are inevitably some issues such as message interception, tampering and other security [1].

In this paper we examine in depth the sinkhole attack, both from the attacker's and defender's point of view. In a sinkhole attack, the attacker tries to attract all traffic through a compromised node, possibly enabling further loopholes. Sinkholes can be created by making the compromised node look very attractive with respect to the routing metrics. The next step towards a complete intrusion detection system for WSNs is to employ resistant countermeasures for defending against an intrusion attempt. Our aim is to illustrate the most effective ways to launch this attack and demonstrate them in practice.

II. SYSTEM MODEL & ASSUMPTIONS

A. Network & Routing Layer Model

In this work, we consider a large set of RPs relying on tree-based topology construction. In this case, data is routed from sensor nodes to the sink through a tree rooted at the sink. In link quality routing protocols, sensor nodes exchange their link quality advertisements to determine good routes to the destination. Two of the most popular RPs falls into this category: the *MintRoute* and the *MultiHopLQI* protocols. MintRoute is used in most real sensor networks deployments today, as for example in [3, 4, 5] and has also served as the basis for the development of the *Collection Tree Protocol* [6]. MultiHopLQI is based on the existence of a hardware indicator, called Link Quality Indicator (LQI), which is believed to be a better indicator of link quality than RSSI. In this paper we investigate one of these protocols, the MultiHopLQI [7]. It has been also used in several sensor network deployments [8]-[10].

B. Threat Model

We assume the presence of an attacker that can access (and eventually change) the internal state of a sensor node. For simplicity, we will assume that the attacker has captured just one node which was previously a legitimate member of the network. To avoid detection, we assume that the attacker does not reprogram the memory of the node, but she rather connects the node to a laptop in order to monitor the packets received.

III. THE SINKHOLE ATTACK

In a Sinkhole attack [2], a compromised node tries to draw all or as much traffic as possible from a particular area, by making itself look attractive to the surrounding nodes with respect to the routing metric. As a result, the adversary manages to attract all traffic that is destined to the base station. In this case, by having the neighboring nodes choose the intruder as their parent, all the traffic coming from their descendants will also end up in the sinkhole. So the attack can be very effective even if it is launched locally, with small effort from the side of the attacker. As these protocols collect network information and decide routing paths periodically, the presence of a sinkhole can compromise the entire network. The strength of this attack stems from its transparency. The malicious node behaves as stated in the protocol and neither performs extra communication nor requires additional hardware.

A. Sinkhole Attack on MintRoute

MintRoute uses link quality estimates as the routing cost metric to build the routing tree toward the base station. For the calculation of these link estimates, MintRoute uses the packet *error rate*. The nodes periodically transmit a packet, called "*route update"* and each node estimates the link quality of its neighbors based on the *packet loss* of the packets received from each corresponding neighbor. Every node maintains a Neighbor Table and updates it when it receives a

route update packet. This table stores a list with the IDs of all neighboring nodes and their corresponding link costs. The node chooses its \parent node" to be the one with the best link quality in the Neighbor Table.

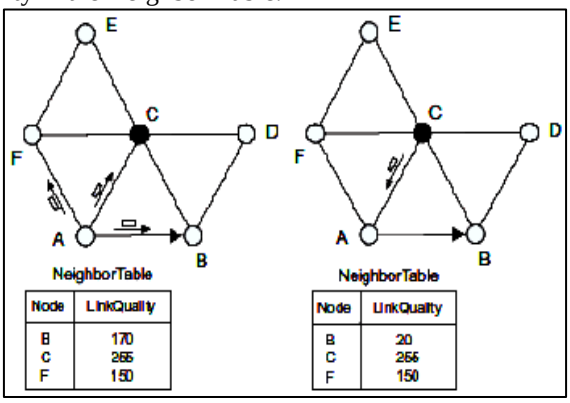

Fig. 1: The two phases of sinkhole attack on MintRoute. (a) Node *C* (attacker) receives the route update packet of node *A*. (b) Node *C* sends a forged packet to *A*, impersonating *B*. In both cases the Neighbor Table of node *A* is indicated.

In the case of a routing protocol, like MintRoute that uses link estimates as the routing metric, the compromised node launching the sinkhole attack will try to persuade its neighbors to change their current parents and choose the sinkhole node as their new one. There are two ways to do that:

1) Advertise an attractive link quality for itself,
2) Make other nodes look like they have worse link quality than itself.

Note that the attacker cannot launch a sinkhole attack by advertising that it has a lower hop count to the base station. Moreover, just advertising a high link quality to the other nodes may not be enough Let's take for example the case shown in Figure 1, where node *C* is the attacker and node *B* is the current parent of node *A*. Node *C* has sent its own route update packet advertising a fake link quality (at the maximum value of 255), but this is not enough to make node *A* change its parent. Therefore, when it receives the route update packet of node *A*, it changes the link quality of node *B* to a low value and sends it back to *A* as a unicast packet, impersonating *B*. Upon receiving this packet, node *A* thinks it is a route update packet from *B*, it extracts the link quality estimation and updates the corresponding entry in the Neighbor Table. This will trigger the parent changing mechanism and since the link quality of node *B* is below 25, that node will be ignored in the selection algorithm and node *C* will be chosen.

B. Sinkhole Attack on MultiHopLQI

The weakness of MintRoute is that each node is based on the advertised link quality from other nodes to decide on its parent. In MultiHopLQI, the nodes calculate the link quality based on their own hardware. Each node periodically broadcasts a *beacon* message and the receivers extract the LQI given by their radio chip. This number is given to a function that calculates the *cost* of the corresponding link. The cost is inversely proportional to the LQI. We will use the notation $Cost_{AB}$ to indicate the cost estimation of node *A* for the link between itself and *B*. The payload of the beacon message includes the sender's current parent and a cost for the whole path to the base station (i.e., the *path cost*). This cost is calculated as the sum of all the costs of the links that make the path. For a node *B* that has a parent *D*, its path cost is calculated as

$$Cost_B = Cost_{BD} + Cost_D \quad \ldots\ldots\ldots\ldots\ldots \quad (I)$$

The value of $Cost_B$ is included in the beacon of node *B*. Node *A* that receives the beacon, reads and stores the value in a table. It also calculates $Cost_{AB}$ as we described above and calculates its own path cost, $Cost_A$, using Equation (I). Node *A* Chooses as its parent the node that minimizes $Cost_A$. According to this algorithm, we identify three ways for an attacker *C* to launch the sinkhole attack:

1) Advertise a low path cost with its parent,
2) Make other nodes look like they have worse path costs than itself,
3) Change its parent to the neighbor with the minimum path cost.

Each of the above strategies using the example shown has been in Figure 2. Let's suppose that initially the nodes have chosen their parents as depicted in Figure 2 (a). The path costs for each node are also indicated. Node *C* is compromised by an attacker and her goal according to the sinkhole attack is to attract as much traffic as possible from the neighboring nodes, convincing them to choose *C* as their parent.

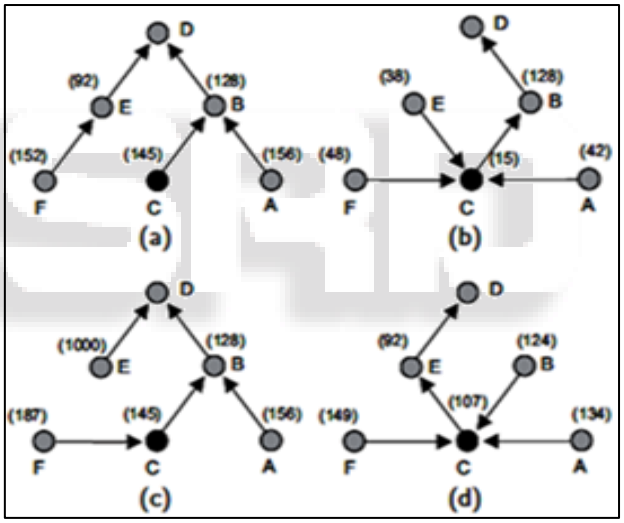

Fig. 2: Three sinkhole attacks on MultiHopLQI. Case (a) shows the original settings of the network before the attack, while cases (b), (c) and (d) show the result of each of the three strategies.

The first and easiest way is to advertise the minimum path cost to the base station. This is shown in Figure 2 (b). According to the function built in MultiHopLQI, the path cost that corresponds to the maximum LQI is 15. The result of this attack is that nodes *A*, *E* and *F* change their parents to *C*, as this reduces their corresponding path costs. This will also trigger the parent changing mechanism at the parent of the attacker, node *B*. However, choosing any of the children of *C* will result in the formation of a routing cycle since *B* is the attacker's parent, and eventually will be forced to go back to its old parent *D*. In the experiments, we noticed that this behavior of *B* kept repeating, however according to the routing protocol, it is legitimate, so we consider that the goal of the attack has been reached. The second way to launch a sinkhole attack is for node *C* to impersonate a node and advertise a very high path cost on its behalf. For example, in

Figure 2 (c), the attacker broadcasts beacons impersonating node *E* and advertises a path cost equal to, let's say, 1000. Its child *F* updates its own path cost to 1000 + $Cost_{EF}$ and realizes that choosing node *C* as its parent will reduce it substantially. Since node *E* will keep broadcasting its legitimate beacons periodically (with path cost 92), the attacker needs to do the same with its spoofed messages, immediately after the messages of *E*. This will keep $Cost_E$ in the memory of node *F* at the attacker's desirable value. If node *C* follows the same strategy for each node in its vicinity, it will manage to attract all the traffic.

The third strategy for the attacker is to look for the node with the minimum path cost in the neighborhood and advertise the best possible, but also legitimate, path cost for itself. For example, in Figure 2 (d), node *E* has the best path cost. In this network, it is the case that

$$Cost_E + Cost_{EC} > Cost_B + Cost_{BC},$$

So node *C* had chosen *B* as its parent. For the attack, however, node *C* chooses *E* as its parent and advertises a very attractive path cost, i.e., $Cost_E$ + 15. This is much less than the path cost it was advertising before. The neighbors will update this value in their tables and hopefully their corresponding path costs will drop by choosing *C* as their parents, as it is the case with Figure 2 (d).

IV. Detecting the Sinkhole Attack

Based on the vulnerabilities of the routing protocols that we propose specific rules that can be used to detect the attack. Since all communication in a WSN is conducted over the air, nodes can listen on the network and capture and examine individual packets passing from their immediate neighborhood in real time.

A. Detection Rules of MintRoute

In order to detect the sinkhole attack on MintRoute, we add a rule that will trigger an alert whenever a malicious node tries to impersonate another node, according to the attack we described in Section 3.1. The intuition is that route update packets should originate only from their legitimate sender and the nodes should defend against impersonation attacks. Detection Rule 1. For each overheard route update packet, check the sender field, which must belong to one of your neighbors.

Figure 3: The estimates of node *A* and node *C* for the quality of the link between them, based on the packet loss rate.

In accordance with *Rule 1*, each node independently measures the link quality estimate of each neighbor and receives their estimates through the route update packets. For example, in Figure 3, we showed the estimate of node *A* for the quality of that link and the estimate of node *C* for the same link and for the same period of time. As it is expected, the estimates of the two nodes for the same link are almost the same, with some small deviation. In particular, the maximum difference that we found between the two link estimates was 49, which corresponds to 19:2%.

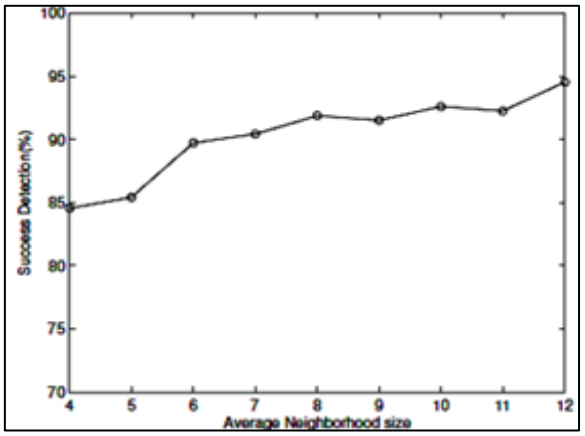

Fig. 4: The Overall Success Rate of Lidea Framework

Detection Rule 2. For each [parent; child] pair of your neighbors, compare the link quality estimate they advertise for the link between them. Their difference cannot exceed 50.

In accordance with *Rule* 2, a monitoring node cannot know which of the two nodes advertises fake link quality. However, because the goal of the attacker is to attract as much traffic as possible *all* of the neighboring IDS watchdogs, enhanced with these detection rules, will be alerted that something is wrong. For example, in Figure 4 we calculated the probability that *LIDeA* system successfully identifies the attacker. To do this we run our intrusion detection protocol for 10:000 different topologies, choosing each time a random attacker. If the voting phase was conclusive the protocol ended; otherwise, the IDS agents were not able to deduce the attacker's identity. As we can see, the protocol always succeeded except for the cases where the topology was such that the intersection of the suspected sets that each agent received produced a set of more than one node. In this case, *s* will be suspected by the same number of nodes as the attacker. Therefore, the voting phase and, hence, the intrusion detection will fail [11]. However, as we can infer from Figure 4, when the network becomes more dense this probability drops, and for more than 7 neighbors in average it becomes less than 10%.

B. Detection Rules on MultiHopLQI

Since some of the attacker's strategies are common between the two routing protocols, the corresponding rules can also be applied to detect the sinkhole attack in MultiHopLQI.

Detection Rule 1 from the previous section can be applied here as well.

For the strategy described in Figure 2 (b), where the attacker advertises the minimum path cost, there is an inconsistency in the protocol itself that we can take advantage and define a new rule. We notice each node should be advertising a bigger path cost than its parent, as it is derived by Equation (I). In this attack, it's not hard to see that this condition is violated.

According to the description of the attack, the attacker advertises a path cost which is smaller that its parent. The nodes that are neighbors of both the attacker and its parent have their path costs stored in their memory, according to the protocol. So they could apply the following rule and detect the attacker: Description Rule 3 for each beacon, check

that the advertised path cost of the node is bigger than the path cost of its father.

If this rule is violated, one of the two nodes lies about its path cost and it has to be the one that advertises the smaller cost. In a different case, the child would immediately update its path cost according to Equation (I) and may trigger the parent changing mechanism, depending on the result.

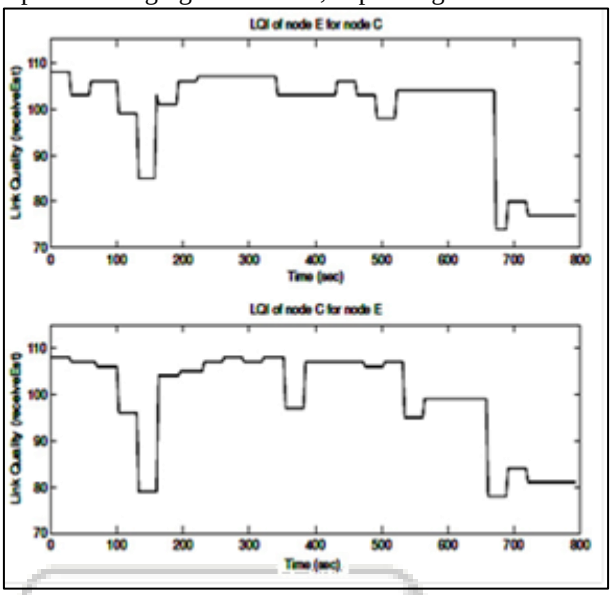

Fig. 5: The estimates of node *E* and node *C* for the quality of the link between them, based on the LQI.

Detecting the third attack that we described in Section 3.2 for MultiHopLQI is more difficult, because as we said, the attacker advertises a path cost that is within the limits and is higher than the cost of its parent, as it is supposed to be. We made the same experiment and compared the LQI of two nodes, *E* and *C*, for the link between them. As shown in Figure 5, they are the same, except for a small deviation. The maximum observed difference was 7. So, for MultiHopLQI, we can define an equivalent rule with *Detection Rule* 2, as follows.

Detection Rule 4. For each [parent; child] pair of your neighbors, compare the LQI they advertise for the link between them. Their difference cannot exceed 10.

The only problem about applying this rule in practice is that nodes in MultiHopLQI do not advertise the LQI that they calculate for their links. We strongly suggest this modification for future designs of similar routing protocols.

V. Existing Sinkhole Countermeasures

The approach presented by Ngai et al: [12] involves the base station in the detection process, resulting in a high communication cost for the protocol. 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.

Another scheme proposed by Choi and Kim [13] can detect a sinkhole attack that uses LQI based routing and several detecting nodes. General nodes collect minimum link costs inside a neighborhood, and detecting nodes compute the minimum path cost with their surrounding detector nodes. Moving on, Tumrongwittayapak et al: [14] present a lightweight and robust solution for detecting the sinkhole and selective forwarding attacks based on RSSI values of received messages. Their proposed scheme needs collaboration of some extra monitor (EM) nodes. They use RSSI values from four EM nodes to determine the positions of all sensor nodes with respect to the base station.

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 Ad-hoc On- demand Distance Vector protocol (AODV) [15] and the Dynamic Source Routing (DSR) protocol [16]. However, in our experience, routing protocols specifically designed for sensor networks, like MintRoute and MultiHopLQI, require much less resources and are usually preferred for such networks.

VI. Conclusions

A sinkhole attack is considered to be a prominent attack that is carried out in order to alter the correct routing in a wireless sensor network. The detection of such an attack is still a significantly challenging task. In this paper, we identified several vulnerabilities of a popular set of routing protocols (*link quality* RPs) for sensor networks and showed how they can be exploited by an attacker for establishing a sinkhole. It turns out that the effort the attacker has to put is minimal and the attack can go undetected, unless certain detection rules are applied. In general, the results of this paper serve a two-fold purpose: they motivate a better design of routing protocols that can make them more resilient to attacks and they also open the way for defining more general and formal rules in intrusion detection designs.

References

[1] B. Karp and H. T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks”, in Proceedings of the 6th annual international conference on Mobile computing and networking, MobiCom '00, (New York, NY, USA), ACM, pp. 243-254, 2000.

[2] C. Karlof and D. Wagner, “Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures”, Ad Hoc Networks, vol. 1, pp. 293-315, Sept. 2003.

[3] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees, “Deploying a wireless sensor network on an active volcano”, IEEE Internet Computing, vol. 10, pp. 18-25, March 2006.

[4] T. Schmid, H. Dubois-Ferriµere, and M. Vetterli, “SensorScope: Experiences with a Wireless Building Monitoring Sensor Network”, in Proceeding of the Workshop on Real-World Wireless Sensor Networks (REALWSN '05), (Stockholm, Sweden), June 2005.

[5] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, “Wireless sensor networks for structural health monitoring”, in SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems, pp. 427-428, 2006.

[6] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, “Collection tree protocol”, in Proceedings of the 7th ACM Conference on Embedded Networked Sensor

Systems, SenSys '09, (New York, NY, USA), ACM, pp. 1-14, 2009.

[7] T.MultiHopLQI.http://www.tinyos.net/tinyos1.x/tos/lib/MultiHopLQI, 2004.

[8] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and yield in a volcano monitoring sensor network", in OSDI '06: Proceedings of the 7th symposium on Operating systems design and implementation, (Berkeley, CA, USA), USENIX Association, 2006.

[9] G. Giorgetti, S. Mastroianni, J. Lewis, G. Manes, and S. Gupta, "The personal sensor network: A user-centric monitoring solution", in BodyNets '07: Proceedings of the 2nd International Conference on Body Area Networks, 2007.

[10] A. Becher, Z. Benenson, and M. Dornseif, "Tampering with motes: Real-world physical attacks on wireless sensor networks", in SPC (J. A. Clark, R. F. Paige, F. Polack, and P. J. Brooke, eds.), vol. 3934 of Lecture Notes in Computer Science, Springer, pp. 104-118, 2006.

[11] J. Paek and R. Govindan, "RCRT: rate-controlled reliable transport for wireless sensor networks", in SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems, (New York, NY, USA), ACM, pp. 305-319, 2007.

[12] E. C. H. Ngai, J. Liu, and M. R. Lyu, "An efficient intruder detection algorithm against sinkhole attacks in wireless sensor networks", Comput. Commun., vol. 30, pp. 2353-2364, Sept. 2007.

[13] B. G. Choi, E. J. Cho, J. H. Kim, C. S. Hong, and J. H. Kim, "A sinkhole attack detection mechanism for lqi based mesh routing in wsn", in Proceedings of the 23rd international conference on Information Networking, ICOIN'09, (Piscataway, NJ, USA), pp. 83-87, 2009.

[14] C. Tumrongwittayapak and R. Varaku- lsiripunth, "Detecting sinkhole attack and selective forwarding attack in wireless sensor networks", in Proceedings of the 7th international conference on Information, communications and signal processing, ICICS'09, (Piscataway, NJ, USA), IEEE Press, pp. 889-893, 2009.

