

Distributed Detection of Wormhole Attacks in Wireless Sensor Networks*

Rennie de Graaf¹, Islam Hegazy^{1,**}, Jeffrey Horton²,
and Reihaneh Safavi-Naini¹

¹ Department of Computer Science, University of Calgary,
Calgary, AB T2N 1N4, Canada

rennie.degraaf@gmail.com, islam.hegazy@ucalgary.ca, rei@ucalgary.ca

² Computer Science and Software Engineering,
University of Wollongong, Wollongong, NSW, 2522 Australia
jhorton@google.com

Abstract. Sensors in a wireless sensor network depend on their neighbours to route their messages. Yet, routing protocols in wireless sensor network are vulnerable to different types of attacks. In this paper, we consider the *wormhole attack* in which the adversary diverts traffic from one part of the network to another part by introducing a low cost tunnel between the two parts. We introduce a distributed intrusion detection system that monitors the communication in the network and propose a criterion for the placement of intrusion detection nodes. The intrusion detection system searches for violations of that criterion to detect wormholes of length above a certain minimum value. We evaluate the effectiveness of our system in a simulated environment. The experiments show that our system can detect 100% of the wormholes that are beyond the communication range of the intrusion detection nodes. Finally, we discuss our results and show directions for future work.

Keywords: Security, Intrusion detection, Wormholes, Wireless sensor networks, Distributed systems.

1 Introduction

A Wireless Sensor Network (WSN) is composed of tiny sensors and one or more sink nodes (base stations). The sensors are responsible for collecting data from their local environments and forwarding it to the sink node(s). These sensors are highly constrained devices: they have limited computation and communication power, and can store only a small amount of data. Securing WSNs is challenging because they can be subjected to a wide range of attacks due to their wireless communication, the insecure environments where they operate, and their limited resources, which preclude the use of resource-intensive security mechanisms.

* This work is in part supported by Informatics Circle of Research Excellence, Alberta, Canada and the Australian Defence Science and Technology Organization, Australia.

** On leave from FCIS, Ain Shams University, Cairo, Egypt for his PhD.

WSNs are susceptible to different types of attacks: physical attacks; resource consumption attacks; and routing disruption attacks [1]. Physical attacks, such as node capturing or traffic jamming, aim to damage the sensor nodes or disable data transmission or steal cryptographic keys to compromise the network. Resource consumption attacks, such as flooding, are achieved by sending bogus data to exhaust the resources of the nodes. Routing attacks aim to disrupt the routing in the network for malicious goals. Routing disruption attacks can be achieved by broadcasting false information, rerouting data to a node other than the original destination, etc.

Routing attacks in WSNs take advantage of the wireless communication and the vulnerability of sensors to being captured. It is difficult to guarantee physical security for the vast areas where WSNs are deployed; this enables adversaries to capture and examine, modify, or remove the sensors, and to insert devices such as malicious nodes or repeaters into the environment. Modified or malicious nodes can be used for selective forwarding [2] and sinkhole attacks [3]; wormhole attacks, where the attacker diverts traffic from one part of the network to another part by introducing a low cost tunnel between the two parts, require modified nodes or repeaters [4]. Sensor network routing protocols, such as AM_ROUTE [5,6], are not developed with security mechanisms against the possible attacks. Protection mechanisms based on cryptographic techniques are not sufficient because the vulnerability of nodes to capture allows the adversary to access cryptographic keys; this enables the adversary to duplicate the sensors or insert new ones. Intrusion Detection Systems (IDSs) provide a complementary means of providing protection. In this paper, we introduce a distributed IDS to detect wormhole attacks in WSNs.

1.1 Related Work

Routing attacks against WSN have gained attention in the literature due to the severity of these attacks, such as black-hole and selective forwarding [2,7], sinkhole [3,8], and sybil attack [9]. Works in detecting wormhole attacks either take a centralized processing approach [10,11] or an in-network processing approach [4,12,13,14]. In the centralized processing approaches, the sensors send network information, e.g., neighbour lists, to the sink, where the detection process is performed. Centralized processing approaches suffer from high energy consumption due to the extra energy required to send the network information to the sink. In the in-network processing approaches, the sensors perform the detection process by monitoring their neighbours or exchanging local network information. In-network processing lacks the global view of the network because the sensors can detect the communication in their neighbourhood only.

Wang et al. [10] propose a centralized processing approach, where a central controller receives distance estimates between the sensors. The central controller then builds the network layout from these estimates and searches for anomalies, which result from the connections through the wormhole. However, distance estimates may be inaccurate, which may hide the existence of a wormhole.

Buttyan et al. [11] propose a centralized approach that assumes that the distribution of sensors is known a priori. The base station builds an estimated distribution graph and compares it using a hypothetical with another graph that it builds from the neighbour lists received from the sensors. If the outcome of the hypothetical test is above a predetermined threshold, then wormholes are detected. This technique is successful if the density of sensors is high and it depends on the prior knowledge of the distribution of sensors.

Hu et al. [12] present an in-network processing approach to detect wormholes. They propose geographical leashes, which require locations of nodes and loosely synchronized clocks between the nodes, and temporal leashes, which require tightly synchronized clocks between the nodes and assume that packets travel at the speed of light. Both approaches are hard to implement in WSNs because locations of nodes and tightly synchronized clocks are not common in WSNs deployments.

Liteworp is another in-network processing approach by Khalil et al. [13]. Liteworp uses one-time authentication to discover neighbours and uses neighbour monitoring to attest their transmissions. Wireless nodes exchange their 1-hop neighbours list with their neighbours to build the 2-hop neighbours list using pair-wise keys. For sensor nodes, saving the neighbour lists and pair-wise keys for each neighbour may not be feasible due to the limited memory resources. In addition, Liteworp does not defend against wormholes that exist before the neighbour discovery process.

Truelink is another in-network processing approach by Eriksson et al. [4]. Truelink extends the MAC layer of IEEE 802.11 to detect wormholes by combining timing and authentication to verify neighbouring nodes. It proceeds in two phases: rendezvous phase and authentication phase. The rendezvous phase requires a strict timing constraint and the authentication phase requires an authentication key. Strict timing may not be achievable in low-cost sensors and it is not clear how the sensors get their authentication keys.

Maheshwari et al. [14] detect wormholes by searching for forbidden structures in the connectivity graphs, which do not occur under normal network operation. The algorithm models each node as a disk of unit radius, then searches for invalid unit disk graphs that are created due to the long distances that the wormholes provide. Each wireless node searches for the forbidden structures in their $2k$ -hop neighbourhood. The accuracy of detecting wormholes using the proposed algorithm is dependent on the density of nodes. In addition, the knowledge of the distribution of nodes is an asset to enhance the detection accuracy and the wireless range of the wireless nodes is not a perfect disk.

1.2 Our Contribution

We introduce a distributed IDS that consists of a number of intrusion detection nodes (ID nodes), which monitor the communications in a WSN. We assume that the ID nodes can communicate securely and do not have the same strict power and computation constraints of the sensors. ID nodes can share their collected data and use this data to detect attacks collaboratively. To the best

of our knowledge, this is the first work that introduces a separate infrastructure to detect routing attacks in WSN. This work overcomes the shortcomings of the centralized and the in-network processing approaches because the sensors do not send any network information either to the sink or to their neighbours. In addition, it does not require tight time synchronization, locations of sensors, nor distribution of sensors.

We consider the application of that architecture to detect wormhole attacks. We first propose a criterion that puts constraints on the placement of ID nodes such that the communication between any two neighbouring sensors is monitored by at least one ID node. Our experiments show that if the ID nodes are placed such that all nodes in the WSN are monitored, wormholes beyond a certain length are detected.

We evaluate our IDS by simulation in *ns-2* [15]. We use different numbers of ID nodes, spaced to satisfy our monitoring criterion, against different wormhole lengths. We evaluate the performance of the IDS in terms of coverage of the ID nodes and detection ratio. In a WSN of 256 sensor nodes, we need 16 ID nodes each with range 24 units to fully cover the WSN if the sensor nodes are deployed in a 16×16 grid. However, for random deployment we need 9 ID nodes each with range 39 units to fully cover the WSN.

The paper is organized as follows. Section 2 provides an overview of the wormholes and their types. In Section 3, we present the architecture of our intrusion detection system and in Section 4 we explain the simulation environment and results. Finally, we conclude the paper in Section 5.

2 Definitions and Models

In a multi-hop WSN, messages travel through multiple sensors to reach the sink. Two sensors form a *valid hop* if they are within the wireless range of each other. A *valid route* is a sequence of valid hops starting from a source and ending at a destination, such that a node appears at most once in the route. In the absence of a secure node authentication mechanism, a route can include malicious nodes that may deviate from the routing protocol in an arbitrary way; for example they may alter or drop messages, or advertise false information.

In a *wormhole* attack, an adversary creates a tunnel between two distant parts of the network and diverts the traffic flow through it [12]. The tunnel provides a low-cost path between the two ends of the tunnel and results in direct connection (one hop) between nodes that would not normally be within the communication range of each other. An adversary can use its control of such a preferential route to launch other attacks such as selective-forwarding. The distance between the two ends of the tunnel is the *length* of the wormhole and is measured by the length of a straight line connecting the two ends. The low cost nature of the out-of-band channel guarantees that a packet entering the wormhole will appear at the other end with low delay.

Two types of wormholes are possible: active wormholes and passive wormholes. In an *active* wormhole, the endpoints of the tunnel are part of the WSN

and take part in the network’s routing protocol. Nearby nodes are likely to route messages through the active wormhole endpoints due to the low-cost route that they offer. In a *passive* wormhole, the endpoints of the tunnel are simple repeaters that intercept, forward, and re-broadcast packets. Nodes near each endpoint are able to hear messages transmitted by nodes near the other end. The nodes will believe that they are immediate neighbours, so they will route packets to distant nodes through the wormhole.

Figure 1 gives examples of active and passive wormholes. In both Fig. 1a and Fig. 1b, node *a* wishes to send a message to node *b*. Nodes *c*, *d* and *e* are other valid hops sensors. Nodes *m* and *n* are wormhole endpoints. Node *a* can establish a valid route to *b* via *c*, *d* and *e* (shown as dotted arrows), but in both cases, it chooses a lower-cost route via the *m-n* wormhole (shown in solid arrows). In the active wormhole case, *m* notifies *a* of its low-cost route to *b*, leading *a* to decide that the cost to *b* via *m* is lower than the cost through *c*. *a*’s hop to *m* is valid, but *m*’s hop through the wormhole to *n* is not, so the route is invalid. In the passive wormhole case, *m* and *n* intercept, relay, and re-broadcast *a*’s and *b*’s transmissions, leading them to believe that they are immediate neighbours. Thus, *a* addresses its messages directly to *b*. Since *a* cannot communicate with *b* directly, the route is invalid.

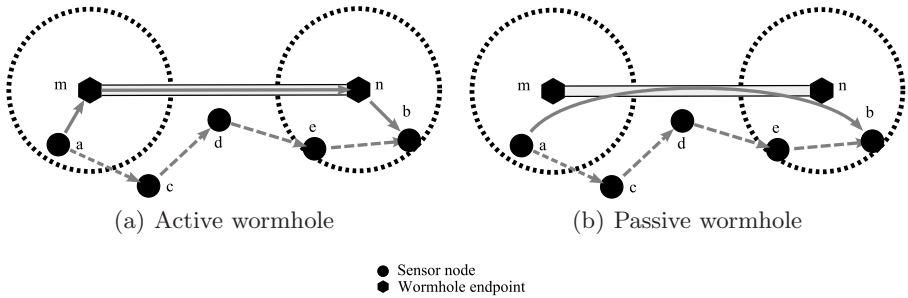


Fig. 1. Active and passive wormholes

2.1 Communication Model

We consider the following model for the sensors and their communication:

1. The network consists of a set S of sensors and a single *sink*. Sensors have unique *ids*, use omnidirectional antennae, and they are static. In the absence of collisions, the sensors can reliably receive messages transmitted by all other sensors within a distance of r_s .
2. Sensors use the AM_ROUTE routing protocol to construct a routing tree rooted at the sink¹. The sink initiates the construction of the tree by broadcasting a beacon, which is forwarded by the sensors. Sensors choose their next hop as the first neighbour from which they hear the beacon. The sink

¹ We base our simulation on the description from the appendix of [5].

periodically broadcasts the beacon for a new tree construction. Sensors initially set their next hop to ∞ , and do not transmit data messages until they identify a route to the sink.

3. We assume that there are no attacks against the MAC protocol.
4. Messages may be dropped in the network due to signal collisions or other factors. This will result in loss of information.

2.2 Adversary Model

We assume that the adversary has physical access to the WSN, so that it can plant the wormhole endpoints. For active wormholes, the adversary needs to know the routing protocol to be able to participate in the routing process as a normal node. For passive wormholes, this knowledge is not required. We also assume that the messages transmitted by the wormhole endpoints cannot be reliably distinguished from messages transmitted by the sensors.

3 Intrusion Detection System

We consider a network of ID nodes that monitor the communication in a WSN and collaborate to detect intrusions. The ID nodes can hear the messages that are sent by the sensors that are within their reception range.

3.1 Intrusion Detection Nodes

We consider the following about the ID nodes:

1. They have unique *ids*.
2. They have larger radio ranges than the sensors, $r_i > r_s$, where r_i and r_s are the communication ranges of the ID nodes and the sensors, respectively.
3. They do not suffer from the limited memory, processing, and power resources as the sensors.
4. They can communicate securely with each other to share their observed data.
5. They have loosely synchronized clocks.

These enhanced capabilities make the ID nodes more expensive than normal sensors, but the cost will be acceptable if they are deployed in small numbers, and if they can provide an effective intrusion detection mechanism. Next we propose a criterion that can be used to determine the number of ID nodes required to monitor all the sensors in the network.

3.2 Detecting Wormholes

ID nodes observe the communication of sensors to infer the relative positions of the sensors. In particular, if an ID node X can hear sensor a and its next hop b , then it can infer that the distance between a and b is at most $2r_i$. Our approach in detecting wormholes uses this observation.

For detecting wormholes, we need a stronger coverage criterion, which we call the *Fully-Monitored Criterion (FMC)*, to be satisfied. This criterion requires the ID nodes to be distributed such that any sensor a and its next valid hop b are monitored by at least one ID node. The distance between any two ID nodes must be chosen such that the criterion is satisfied. In other words, the number and positions of ID nodes are chosen such that all pairs of valid hop sensors in the network are monitored by at least one ID node.

Definition 1. *A sensor s is fully-monitored by an ID node X if both s and its next valid hop are within the reception range X . A WSN satisfies the fully-monitored criterion (FMC) if for all sensors $s \in S$ and all valid routes, there exists some ID node X that fully-monitors s .*

We detect wormholes by searching for violations of the FMC. Using a network of ID nodes, we can detect active and passive wormholes using *FMC Validation*, and can distinguish between passive and active wormholes using *Wormhole length Check*.

FMC Validation. Assuming that the fully-monitored criterion holds, an ID node X can hear all the sensors that are within distance r_i . This means that two sensors that are fully-monitored by X must be separated by a distance at most $2r_i$. If ID node X can hear sensor a but not its next hop b , then X assumes that a is fully-monitored by a neighbour ID node Y with an overlapping coverage area.

A WSN will be fully-monitored if the ID nodes monitor all sensors such that any pair of sensors that form a valid hop is monitored by at least on ID node. For the FMC to hold, the communication ranges of the ID nodes must overlap to monitor the pair of valid hop sensors that are not in the same communication range of a single ID node. Specifically, the diameter of the overlapping area between 4 ID nodes must be at least equal to the communication range of sensors as shown in Fig. 2. Sensors a and b are placed at the boundary of the communication range, r_s , of each other; if the overlapping coverage area between ID nodes W , X , Y and Z is less than r_s , then the communication between a and b will be detected as a wormhole because none of the ID nodes can fully-monitor a or b .

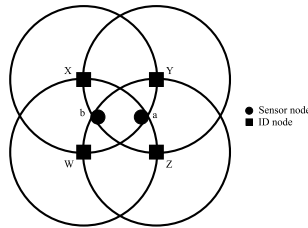


Fig. 2. Intersection of ID nodes

Active Wormholes. The endpoints of an active wormhole are two sensors that participate in the routing protocol and they route packets with their own *ids*. However, they are not distinguishable from the other normal sensors. Thus, any active wormhole of length greater than $2r_i$ will be detected because no ID node can fully-monitor the two endpoints of the wormhole. If an ID node hears a sensor but not its next hop, wormhole endpoints in this case, then it will query the neighbouring ID nodes to know if any of them fully-monitors that endpoint. If none of the neighbouring ID nodes fully-monitors that sensor, then an endpoint of the wormhole is detected.

Figure 3 shows an example of sensor *a* routing through an active wormhole with endpoints *m* and *n* in the presence of ID nodes *X*, *Y* and *Z*. Since *X* can hear both *a* and *m*, it fully-monitors *a*. Likewise, *Y* hears both *n* and *b*, so it fully-monitors *n*. *Z* fully-monitors *b*. With all the sensors fully-monitored either by itself or by a neighbour, *Y* detects no wormholes. However, *X* does not fully-monitor *m*, and neither does its neighbour, *Y*, so *X* detects *m* as a wormhole endpoint.

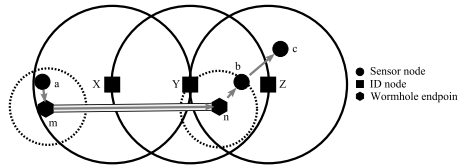


Fig. 3. Detecting a wormhole using FMC validation

Some wormholes may have length $r_s < l \leq 2r_i$; in this case the wormhole will not always be detected as both nodes may be located within the reception range of a single ID node.

Passive Wormholes. The endpoints of a passive wormhole do not participate in the routing protocol and are not visible to the sensors and the ID nodes. The wormhole intercepts packets from sensors near one endpoint and sends them to the sensors near the other endpoint. Thus, packets passing through the wormhole have the sender and receiver *ids* of legitimate sensors. This makes detection of passive wormholes harder. In particular, the length criterion used for detection of active wormholes will not work. For example, if the wormhole in Fig. 3 is passive, then *X* perceives *a* and *b* within its reception range; *Y* reaches a similar conclusion and so both *X* and *Y* believe that they fully-monitor *a*. *Z* fully-monitors *b*, so *Y*, which is a neighbour of *Z*, does not detect *b* as a wormhole endpoint. However, *X* also hears *b*, and it is *not* a neighbour of *Z*, so it detects *b* as a wormhole. This behaviour may not always occur; if *c* is within the reception range of *Y*, then *Y* will fully-monitor *b* and no ID node can detect the wormhole.

Using the FMC, there is a minimum length for the passive wormhole tunnel so that the ID nodes can detect the wormhole. To be fully detected, the distance between any two sensors communicating through the passive wormhole must be greater than three times the communication range of the ID nodes. Otherwise, the wormhole may not be detected. In Fig. 4, three ID nodes $X, Y,$ and Z monitor three sensors $a, b,$ and c . There exists a passive wormhole with endpoints m and n . Suppose that a and b communicate through the wormhole and they are placed at the boundary of the communication ranges of the wormhole endpoints m and n , respectively. ID node X can hear a communicating with b and it can hear b , so X fully-monitors a . However, X cannot fully-monitor b because it cannot hear c . Since b is not in the range of Y , then Y does not fully-monitor b . The wormhole will be detected because neither X nor its neighbour Y can fully-monitor b . However, if b is in the range of Y , the wormhole will not be detected because Y can fully-monitor b .

As in active wormholes, sensors communicating over shorter distances through a passive wormhole may also be detected, but this cannot be guaranteed.

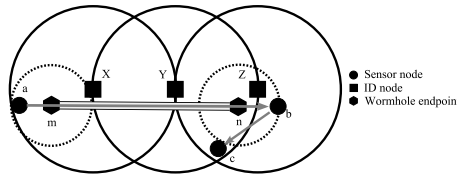


Fig. 4. Minimum length between 2 nodes communicating through a passive wormhole

Distinguishing Active from Passive Wormholes. FMC Validation can detect active and passive wormholes (with different lower bounds on their lengths) but cannot distinguish between the two cases. It is important to make this distinction to be able to provide an appropriate response.

To achieve this distinction, we again take advantage of the fully-monitored assumption, as well as the message relaying property of passive wormholes. This property will cause a node to be fully-monitored by two ID nodes that are not neighbours: if a passive wormhole is detected at some legitimate sensor s , then s must be fully-monitored by at least two ID nodes that are *not* neighbours. Conversely, if a sensor s is fully-monitored by two or more ID nodes that are not neighbours, then its messages must be repeated through a wormhole. To check for this condition, ID nodes can share their lists of fully-monitored sensors with all other ID nodes that are not neighbours, and can then compare the lists from other ID nodes with their own list to identify the sensors that are near to passive wormhole endpoints. This result can then be compared with the list of wormholes detected to distinguish passive from active wormholes.

Algorithm 1 describes the steps of the system. The system runs for a period of time *run_time*. It first begins by initializing the sensors and the ID nodes,

then it enters a loop until *run_time* expires. The sink broadcasts a beacon every *route_update* period. If a sensor hears a beacon, it will update its next hop and rebroadcast the beacon. Every sensor sends a data packet to its next hop every *send_time* period, and its next hop forwards this data packet to its next hop until the data packet reaches the sink. At the same time, the ID nodes monitor the communication in the WSN and check for wormholes every *check_interval* period. The subroutines of the ID nodes are described in Algorithm 2. When an ID node starts, it runs the INIT procedure to initialize the data structures. When a message from a sensor is heard, an ID node runs the RECV procedure to incorporate any useful information from that message into its route tables. Finally, all ID nodes periodically run the CHECK procedure, in which they identify the sets of sensors that they hear and fully-monitor, and share this information with other ID nodes to identify wormholes.

Algorithm 1. Main system algorithm

```

Initialize sensors                                     ▷ Set next hop to  $\infty$  in all sensors
ID nodes call INIT()
while run_time not expired do
  if route_update expired then
    Sink broadcasts a beacon packet
  if sensor a receives the beacon then
    a updates its next hop and rebroadcasts the beacon packet
  if sensor a send_time expired then
    a sends a data packet to its next hop
  if sensor a receives a data packet then
    a forwards data packet to its next hop
  if ID node i hears a packet m then
    i calls RECV(m)
  if ID node i check_interval expired then
    i calls CHECK()
  
```

3.3 False Alarms

As previously mentioned, active wormholes shorter than $2r_i$ and passive wormholes shorter than $3r_i$ may not be detected; these may be viewed as *false negative* results. *False positives* may also occur in the following cases:

1. *Lack of information due to collisions:* For ID nodes *X* and *Y* that monitor sensors *a*, *b*, and *c* as in Fig. 5, suppose that the transmission sequence is $a \rightarrow b \rightarrow c$. ID node *X* fully-monitors *a* because *a* and *b* are in its communication range. If, due to collision, ID node *Y* does not hear the communication between *b* and *c*, then *Y* does not fully monitor *b*. When ID nodes *X* and *Y* exchange their lists, ID node *X* will not find *b* in the list of ID node *Y* so it will assume that *b* is a wormhole.

Algorithm 2. IDS subroutines

```

procedure INIT():                                ▷ Run when ID node  $i$  starts
  set  $S \leftarrow$  the set of all sensors
  set  $I \leftarrow$  the set of all ID nodes
  set  $N \leftarrow$  all ID nodes whose reception areas intersect with  $i$ 's reception area
  for each  $s$  in  $S$  do
    set  $route[s] \leftarrow \alpha$                 ▷ The most recent route for  $s$ 

procedure RECV( $m$ ):                               ▷ Run when ID node  $i$  intercepts a packet
  if  $m$  is a data packet or a beacon packet then
    set  $route[m.sender] \leftarrow m.nexthop$ 

procedure CHECK():                                ▷ Run periodically by ID node  $i$ 
  set  $F \leftarrow \emptyset$                         ▷ Nodes fully-monitored by  $i$ 
  set  $P \leftarrow \emptyset$                         ▷ Nodes heard but not fully-monitored by  $i$ 
  set  $D \leftarrow \emptyset$                         ▷ Nodes fully-monitored by non-neighbouring ID nodes
  for each  $s$  in  $S$  do
    if  $route[s] \neq \alpha$  then
      if  $route[route[s]] \neq \alpha$  then
        set  $F \leftarrow F \cup \{s\}$                 ▷  $s$  is fully covered
      else
        set  $P \leftarrow P \cup \{s\}$                 ▷  $s$  is partially covered
  for each  $i$  in  $I$  do
    send  $F$  to  $i$ 
  set  $P' \leftarrow P$                             ▷ The set of possible local wormhole nodes
  for each  $i$  in  $I$  do
    receive  $F_i$  from  $i$ 
    if  $i \in N$  then
      set  $P' \leftarrow P' \setminus F_i$ 
    else
      set  $D \leftarrow D \cup F_i$ 
  for each  $s$  in  $P'$  do
    if  $s \in D$  then
      output  $s$                                     ▷  $s$  is a passive wormhole
    else
      output  $s$                                     ▷  $s$  is an active wormhole

```

2. *Inconsistent views due to unsynchronized clocks:* Suppose that after ID nodes X and Y exchange their fully-monitored lists, Y fully-monitors a as in Fig. 6a. Suppose that a new beacon is broadcast and a changed its next hop to b as in Fig. 6b. Now, X fully-monitors a . If X and Y are not synchronized, Y may check for wormholes without receiving the new fully-monitored list from X so it will detect a as a wormhole.
3. *Lack of sink information:* since the sink does not forward messages, it may be identified as a wormhole. To avoid this, ID nodes must be aware of the *id* of the sink and treat it as a special case.

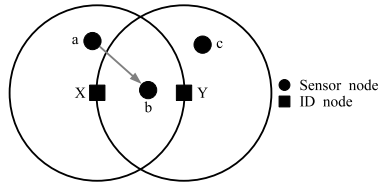


Fig. 5. Lack of information of Y

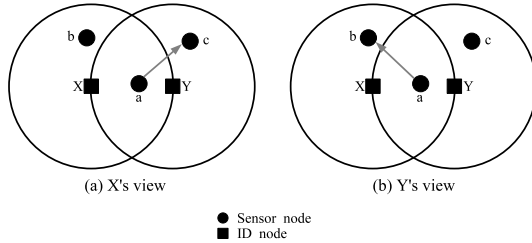


Fig. 6. Inconsistency between ID nodes views

3.4 Response to Wormholes

When an ID node discovers an active wormhole entry point, it can broadcast a message to the sensors within its communication range directing them to blacklist the wormhole node; this will neutralize the wormhole.

Responding to a passive wormhole is more complicated since the node identified as the wormhole is not the actual endpoint. To close a passive wormhole, all the nodes within the communication range of one or both endpoints must be blacklisted; identifying all of them will require multiple cycles of detection, blacklisting, and re-routing. Since false positive detections will also be blacklisted, nodes should expire from the blacklist after a period of time. This may allow routes to be re-established through real wormholes, but these should be detected and closed again after the next detection cycle.

Automated intrusion response has the potential to cause severe collateral damage. None of the nodes blacklisted when attempting to close a passive wormhole actually have anything to do with the adversary. Blacklisting large blocks of sensors may severely limit the routes available in the WSN. In the worst case, closing a wormhole may result in parts of the network that formerly routed through it becoming disconnected from the sink. The costs of allowing messages to flow through the wormhole must be weighed against the costs of collateral damage when deciding whether to use such an automated response system.

4 Simulation

To test our IDS, we simulated a WSN using *ns-2*. The WSN comprised 256 sensors in grid and random deployments in a square environment, 100 units on each side. We conducted the tests using the AM_ROUTE routing protocol. The sink broadcasts a beacon every 10 time units. Sensors generate data messages at uniform random intervals with a mean of 2 time units. The sink was located at the northwest corner of the simulation environment. The ID nodes were positioned using a grid topology.

For grid-topology tests, the sensors had a communication range of 8 units and were positioned in an evenly-spaced 16×16 grid; this allowed each sensor to communicate with the sensors to the north, south, east, and west. For random-topology tests, the sensors were placed with a uniform random distribution with a communication range of 13 units.

We ran a number of tests with the following objectives:

1. Validating the strength of the ID nodes in detecting active and passive worm-holes and in particular measuring the detection ratio.
2. Investigating the trade-off between system parameters and in particular the number of ID nodes and the detection strength of the system.

4.1 Connectivity of Sensor Nodes

For a grid topology, we calculated the minimum communication range of the sensors manually. To have full connectivity, each sensor should have a communication range of at least 6.67 units.

For random topologies, we ran the simulations 1000 times for each of the following sensor ranges; 8, 9, 10, 11, 12, 13, 14, 15, and 16. To fully connect the WSN, the sensors should have a communication range of at least 16 units. However, sensors of communication range 13 units give more than 95% connectivity as shown in Fig. 7.

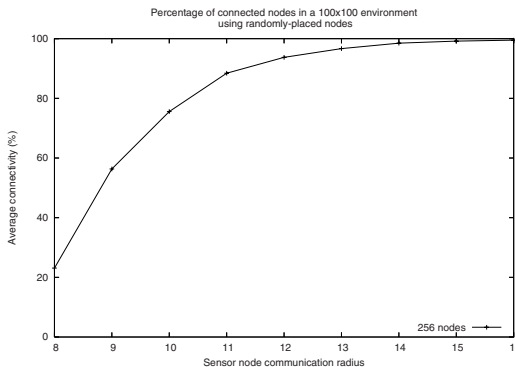


Fig. 7. Percentage of randomly-placed sensors with a valid route to the sink

4.2 ID Nodes Density

We consider ID nodes deployed in grid topologies in our experiments. The number and range of the ID nodes were determined experimentally with the aim of satisfying the FMC. Table 1 summarizes the number and ranges of ID nodes in two WSN deployments. The sensors had a communication range of 8 units in the first deployment and a communication range of 13 units in the second deployment. We can see that we need 16 ID nodes with a communication range of 24 units if the sensors have a communication range of 8 units. If the sensors have a communication range of 13 units, we will need 9 ID nodes with a communication range of 39 units.

Table 1. ID nodes required for full network coverage using a grid deployment

(a) Sensor range = 8		(b) Sensor range = 13	
ID node range	Required ID nodes	ID node range	Required ID nodes
12	81	19.5	25
16	36	26.0	16
20	25	32.5	9
24	16	39.0	9

4.3 Effect of Wormhole Length on Detection

To measure the success of our IDS to detect wormholes, we simulated different wormhole lengths against different ID node ranges. We ran the simulation 100 times for each combination of wormhole length and ID nodes range, then we took the average of the detection ratio. We simulated wormholes with lengths: 8; 16; 24; 32; 40; 48; 56, and 64 units against ID nodes with ranges: 16; 20; 24; 28 and 32 units.

For active wormholes, Fig. 8 shows the results of the simulations in grid and random deployments of sensors. We can see that the system fully detects the active wormholes if the wormhole length is two times greater than the communication range of the ID nodes. For example, ID nodes, with a communication range of 16 units, have a detection ratio about 100% for wormholes with length greater than 32 units.

Figure 9 shows the results of simulating the different passive wormholes lengths in grid and random deployments of sensors. We can see that the results confirm that the passive wormhole will be detected if the length between two communicating nodes through the wormhole is three times greater than the communication range of the ID nodes. For example, in the grid deployment, Fig. 9(a), ID nodes, with a communication range of 20 units, have a detection ratio about 100% for wormholes with length greater than 48 units. If two sensors are communicating through the passive wormhole and they are placed at the communication boundaries of the wormhole endpoints, then $48+16 > 20 \times 3$.

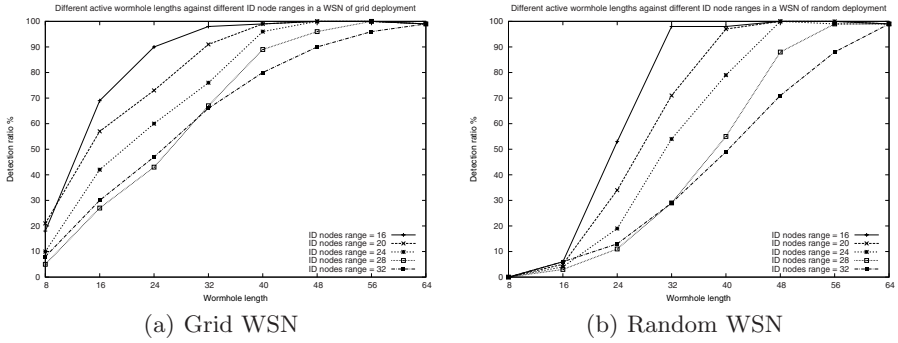


Fig. 8. Active wormhole detection ratio

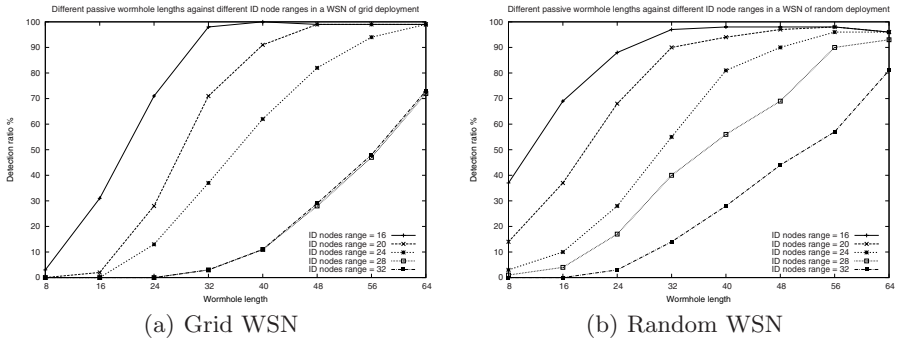


Fig. 9. Passive wormhole detection ratio

5 Conclusions and Future Work

In this paper, we introduced a distributed IDS to detect wormhole attacks in WSNs. We proposed the fully-monitored criterion that requires at least one ID node to monitor the communication between any pair of valid hop sensors. We showed that wormholes can be detected if the fully-monitored criterion is violated. The detection ratio of our system is about 100% if the wormhole connects a pair of sensors that is not monitored by one ID node. In contrast to other wormhole detection mechanisms in WSNs, the detection criterion of our system does not depend on network information, such as the locations of the sensors. In addition, the knowledge of the distribution or the density of the sensors is not required. Using this IDS architecture for detecting other routing attacks is an interesting direction for future research. Also, we will work on enhancing the detection capability of the current system for shorter wormholes.

References

1. Patwardhan, A., Parker, J., Joshi, A., Jorga, M., Karygiannis, T.: Secure Routing and Intrusion Detection in Ad Hoc Networks. In: 3rd IEEE International Conference on Pervasive Computing and Communications, pp. 191–199. IEEE Press, Los Alamitos (2005)
2. Yu, B., Xiao, B.: Detecting Selective Forwarding Attacks in Wireless Sensor Networks. In: 20th International Parallel and Distributed Processing Symposium, pp. 8–15. IEEE Press, Los Alamitos (2006)
3. Ngai, E.C.H., Lie, J., Lyu, M.R.: On the Intruder Detection for Sinkhole Attack in Wireless Sensor Networks. In: IEEE International Conference on Communications, pp. 3383–3389. IEEE Press, Los Alamitos (2006)
4. Eriksson, J., Krishnamurthy, S.V., Faloutsos, M.: Truelink: A Practical Countermeasure to the Wormhole Attack in Wireless Networks. In: 14th IEEE International Conference on Network Protocols, pp. 75–84. IEEE Press, Los Alamitos (2006)
5. Hill, J.: A Software Architecture Supporting Networked Sensors. Master's thesis, Dept. of Electrical Eng. and Computer Science. University of California at Berkeley (2000)
6. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Vuller, D., Pister, K.: System Architecture Directions for Networked Sensors. In: 9th International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 93–104 (2000)
7. Ioannis, K., Dimitriou, T., Freiling, F.C.: Towards Intrusion Detection in Wireless Sensor Networks. In: 13th European Wireless Conference (2007)
8. Krontiris, I., Dimitriou, T., Giannetsos, T., Mpasoukos, M.: Intrusion Detection of Sinkhole Attacks in Wireless Sensor Networks. In: Kutyłowski, M., Cichoń, J., Kubiak, P. (eds.) ALGOSENSORS 2007. LNCS, vol. 4837, pp. 150–161. Springer, Heidelberg (2008)
9. Newsome, J., Shi, E., Song, D., Perrig, A.: The Sybil Attack in Sensor Networks: Analysis & Defenses. In: 3rd International Symposium on Information Processing in Sensor Networks, pp. 259–268. ACM, New York (2004)
10. Wang, W., Bhargava, B.: Visualization of Wormholes in Sensor Networks. In: ACM Workshop on Wireless Security, pp. 51–60. ACM, New York (2004)
11. Buttyán, L., Dora, L., Vajda, I.: Statistical Wormhole Detection in Sensor Networks. In: 2nd European Workshop on Security and Privacy in Ad Hoc and Sensor Networks, pp. 128–141. Springer, Berlin (2005)
12. Hu, Y.C., Perrig, A., Johnson, D.B.: Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks. In: 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 1976–1986. IEEE Press, Los Alamitos (2003)
13. Khalil, I., Bagchi, S., Shroff, N.B.: LITEWORP: A Lightweight Countermeasure for the Wormhole Attack in Multihop Wireless Networks. In: International Conference on Dependable Systems and Networks, pp. 612–621. IEEE Computer Society, Washington (2005)
14. Maheshwari, R., Gao, J., Das, S.R.: Detecting Wormhole Attacks in Wireless Networks Using Connectivity Information. In: 26th IEEE International Conference on Computer Communications, pp. 107–115. IEEE Press, Anchorage (2007)
15. The network simulator, <http://www.isi.edu/nsnam/ns>