Chapter 19

DETECTING WORMHOLE ATTACKS IN WIRELESS SENSOR NETWORKS

Yurong Xu, Guanling Chen, James Ford and Fillia Makedon

Abstract Wormhole attacks can destabilize or disable wireless sensor networks. In a typical wormhole attack, the attacker receives packets at one point in the network, forwards them through a wired or wireless link with less latency than the network links, and relays them to another point in the network. This paper describes a distributed wormhole detection algorithm for wireless sensor networks, which detects wormholes based on the distortions they create in a network. Since wormhole attacks are passive in nature, the algorithm uses a hop counting technique as a probe procedure, reconstructs local maps for each node, and then uses a "diameter" feature to detect abnormalities caused by wormholes. The main advantage of the algorithm is that it provides the locations of wormholes, which is useful for implementing countermeasures. Simulation results show that the algorithm has low false detection and false toleration rates.

Keywords: Wireless sensor networks, wormhole detection, distributed algorithm

1. Introduction

Wireless sensor networks (WSNs) [1, 15] are constructed using numerous small, low-power devices that integrate limited computation, sensing and radio communication capabilities. They provide flexible infrastructures for numerous applications, including healthcare, industry automation, surveillance and defense.

Currently, most WSN applications are designed to operate in trusted environments. However, security issues are a major concern when WSNs are deployed in untrusted environments. An adversary may disable a WSN by interfering with intra-network packet transmission via wormhole attacks, sybil attacks [11], jamming or packet injection attacks [17].

This paper focuses on wormhole attacks [2, 6, 12]. In a typical wormhole attack, the attacker receives packets at one point in the network, forwards them

Xu, Y., Chen, G., Ford, J. and Makedon, F., 2008, in IFIP International Federation for Information Processing, Volume 253, Critical Infrastructure Protection, eds. E. Goetz and S. Shenoi; (Boston: Springer), pp. 267–279.

through a wired or wireless link with less latency than the network links, and relays the packets to another point in the network. Such an attack does not require any cryptographic knowledge; consequently, it puts the attacker in a powerful position compared with other attacks (e.g., sybil attacks and packet injection attacks), which exploit vulnerabilities in the network infrastructure. Indeed, a wormhole attack is feasible even when the network infrastructure provides confidentiality and authenticity and the attacker does not have the cryptographic keys.

Several methods have been proposed for detecting wormhole attacks. However, these methods usually require that some nodes in the network be equipped with special hardware. Solutions such as SECTOR [2] and Packet Leashes [6] need time synchronization or highly accurate clocks to detect wormholes. The method of Hu and Evans [4] requires a directional antenna to be deployed at each node. LAD [3], SerLoc [8] and the approach of Hu, Perrig and Johnson [5] involve anchor nodes (nodes that know their exact locations), which requires the manual setup of a network.

This paper presents the wormhole geographic distributed detection (WGDD) algorithm, which requires neither anchor nodes nor any additional hardware. Since a wormhole attack is passive, the algorithm employs a hop counting technique as a probe procedure, reconstructs local maps using multidimensional scaling at each node, and uses a novel "diameter" feature to detect distortions produced by wormholes. The principal advantage of the algorithm is that it provides the approximate location of a wormhole, which can assist in implementing defense mechanisms. Simulation results show that the wormhole detection algorithm has low false detection and false toleration rates.

2. Related Work

Early approaches proposed for detecting wormhole attacks in wireless ad hoc networks include Packet Leashes [6] and SECTOR [2], which employ the notions of geographical and temporal leashes. The assumption is that each network node knows its exact location, and embeds the location and a timestamp in each packet it sends. If the network is synchronized, then any node that receives these packets can detect a wormhole based on differences in the observed locations and/or calculated times. Such a solution requires a synchronized clock and each node to know its location. The algorithm proposed in this paper does not have these requirements.

Kong, *et al.* [7] have studied denial-of-service (DoS) attacks (including wormhole attacks) on underwater sensor networks. Because these networks typically use acoustic methods to propagate messages, the detection techniques cannot be applied directly to wireless sensor networks.

Hu and Evans [4] have attempted to detect wormholes by equipping network nodes with directional antennas so they can all have the same orientation. Lazos and Poovendran [8] have applied a similar idea in their secure localization scheme called SeRLoc. SeRLoc employs about 400 anchor nodes (called "beacon nodes") in a 5,000-node network. Each anchor node has a directional

antenna and knows its physical location. Other nodes in the network use anchor nodes to locate themselves. Since a wormhole produces shortcuts in a network, the directional antennas deployed at anchor nodes help detect the attack; nodes can then defend against the attack by discarding incorrect localization messages. However, SeRLoc is unable to detect wormhole attacks when anchor nodes are compromised, especially nodes located near one of the ends of a wormhole.

More recently, Liu, *et al.* [3, 9] have proposed an anchor-based scheme for detecting several attacks, including wormhole attacks. The scheme uses a hop counting technique to estimate the distance between a node and an anchor node (called a "location reference"). Since a wormhole changes the distance from a node to an anchor node, a simple threshold method can be used to determine if the change in distance is caused by a wormhole or by a localization error. Our approach also uses a hop counting technique, but it does not involve anchor nodes and, consequently, does not require the manual setup of a sensor network.

A graph-theoretic framework has also been proposed for detecting wormhole attacks [13]. However, the framework relies on "guard nodes," which are functionally similar to anchor nodes.

MDS-VOW [16] provides visualization facilities to detect the distortions produced by a wormhole in a computed network map. The principal limitation of MDS-VOW is that it is a centralized approach; also, as noted in [13], MDS-VOW cannot be applied to networks with irregular shapes. Our approach also detects wormholes based on the distortions they produce in a network map, but it employs a distributed algorithm and can detect wormholes in irregularly-shaped networks.

3. Wormhole Attacks

In a typical wormhole attack, the attacker receives packets at one point in the network, forwards them through a wireless or wired link with much less latency than the default links used by the network, and then relays them to another location in the network. In this paper, we assume that a wormhole is bi-directional with two endpoints, although multi-end wormholes are possible in theory.

A wormhole receives a message at its "origin end" and transmits it at its "destination end." Note that the designation of wormhole ends as origin and destination is dependent on the context. We also assume a wormhole is passive (i.e., it does not send a message without receiving an inbound message) and static (i.e., it does not change its location).

4. Wormhole Detection Algorithm

Our wormhole geographic distributed detection (WGDD) algorithm uses a hop counting technique as a probe procedure. After running the probe procedure, each network node collects the set of hop counts of its neighbor nodes that are within one/k hops from it. (The hop count is the minimum number of

node-to-node transmissions to reach the node from a bootstrap node.) Next, the node runs Dijkstra's (or an equivalent) algorithm to obtain the shortest path for each pair of nodes, and reconstructs a local map using multidimensional scaling (MDS). Finally, a "diameter" feature is used to detect wormholes by identifying distortions in local maps.

The main steps involved in the wormhole detection algorithm are described in the following sections.

4.1 Probe Procedure

Since a wormhole attack is passive, it can only occur when a message is being transmitted in the region near a wormhole. To detect a wormhole attack, we use a probe procedure that floods the network with messages from a bootstrap node to enable all network nodes to count the hop distance from themselves to the bootstrap node. The probe procedure is based on the hop coordinates technique [18].

- **Bootstrap Node:** The bootstrap node x creates a probe message with $(i = id_x)$ to flood the network. Next, the bootstrap node drops all probe messages that originated from itself. The bootstrap node has the hop coordinate $hop_x = 0$ and $offset_x = 0$.

- **Other Nodes:** The probe procedure is presented in Algorithm 1. The algorithm computes the hop distance for node a. Node b is a neighbor of node a; hop_a is the minimum number of hops to reach node a from the bootstrap node (x) and its initial value is MAXINT. The combination of hop_a and $offset_a$ is the hop coordinate for node a. N_a is the set of nodes that can be reached from node a in one hop, and $|N_a|$ is the number of nodes in N_a.

4.2 Local Map Computation Procedure

In this step, each node computes a local map for its neighbors based on the hop coordinates computed in the previous step. After the hop coordinates are generated by the probe procedure, each node requests its neighbor nodes that are within one/k hops to send it their hop coordinates.

After a node receives the hop coordinates from its neighbors, it computes the shortest paths between all pairs of nodes one/k hops away using Dijkstra's algorithm (or a similar algorithm).

Next, multidimensional scaling (MDS) is applied to the $(|N_a|+1 \times |N_a|+1)$ shortest path matrix to retain the first two (or three) largest eigenvalues and eigenvectors for constructing a 2-D (or 3-D) local map. Note that $|N_a|$ is the number of nodes that can be reached from node a in one/k hops.

This step has a computational cost of $O(|N_a|^3 n)$ and a memory cost of $O(|N_a|^2)$ per node. No communication cost is associated with this step.

Algorithm 1 : Probe Procedure (for Node a).

1: INPUT: message (hop_b) from node b $\in N_a$
2: **for** message (hop_b) from any B $\in N_a$ and not TIMEOUT **do**
3: **if** $hop_b < hop_a$ **then**
4: $hop_a = hop_b + 1$
5: forward (message (hop_a)) to MAC
6: **else**
7: drop (message (hop_b))
8: **end if**
9: **end for**
10: **if** $|N_a| == 0$ **then**
11: $offset_a = 0$
12: **else**
13: $offset_a = \frac{\sum_{b \in N_a}(hop_b - (hop_a - 1)) + 1}{2(|N_a| + 1)}$
14: **end if**
15: **return** hop_a and $offset_a$

4.3 Detection Procedure

The detection procedure uses a local map created in the previous step. To help clarify the methodology, we examine the effect of a wormhole on a computed map.

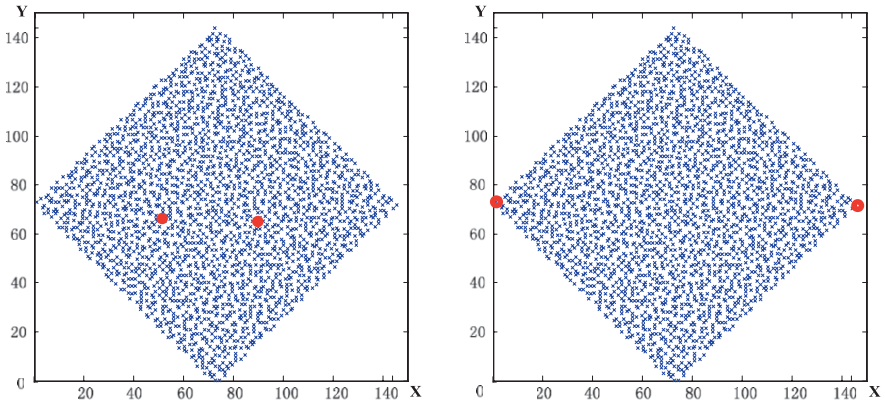
Wormhole in a Reconstructed Map In order to observe a wormhole, we implemented the probe procedure and the local map computation procedure as routing agents, and the bootstrap node for the probe procedure as a protocol agent in ns-2 version 2.29 [10]. The RF range was 15 m.

The first experiment used 2,500 nodes in a uniform placement. Specifically, 2,500 nodes were placed on a grid with $\pm 0.5r$ randomized placement error, where $r = 2$ m is the width of a grid square. A wormhole was implemented as a wired connection.

Figure 1 shows two views of the sensor network. Each "x" mark represents a node; the circles indicate wormhole ends. The wormhole in Figure 1(a) is located in the center of the network. The two ends of the wormhole in Figure 1(b) are at the edges of the network.

Feature for Detecting Wormhole Attacks Since each node has limited resources and cannot store global information, a node can only use local information to detect wormhole attacks.

Figure 2 shows the portions of the network in the vicinity of the two ends of the wormhole in Figure 1(a). Nodes are represented by "x" marks and triangles; dotted circles are used to represent the neighborhoods corresponding to the circled node's transmission range R. After the circled node has completed its computations for the nodes in its local range, it generates the local map shown



(a) Original locations of wormhole. (b) Wormhole at network edges.

Figure 1. A 2,500-node network ($r = 2$ m) with one wormhole.

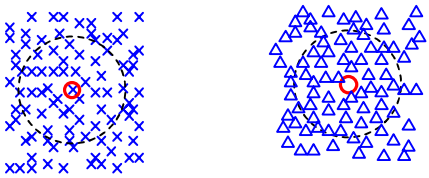


Figure 2. Portions of the network near the wormhole ends ($r = 4$ m; $R = 15$ m).

in Figure 3. The figure shows that, because the wormhole shortcuts the two portions of the network, the circled node can reach farther than before (the longest distance in the local map is 49 m), although the computed local map is distorted by the wormhole.

Based on the above observation, we employ the diameter of the computed local map as a feature to detect wormholes. We define the diameter d for a node a as:

$$d = max(distance(b, c))/2 \tag{1}$$

where $b, c \in N_a$. Note that N_a is the set of neighbor nodes of node a, and $distance(a, b)$ in the 2-D case is computed as $\sqrt{((x - x')^2 + (y - y')^2)}$, where (x, y) and (x', y') are the coordinates of nodes a and b, respectively, in the local map computed in the previous step.

In theory, the diameter of the neighborhood of a node is no more than R because a node can only hear from its neighbors within the transmission range R. However, the "shortcuts" created by a wormhole distort the computed map in the neighborhood of a node. Therefore, the diameter of the computed local

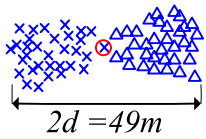
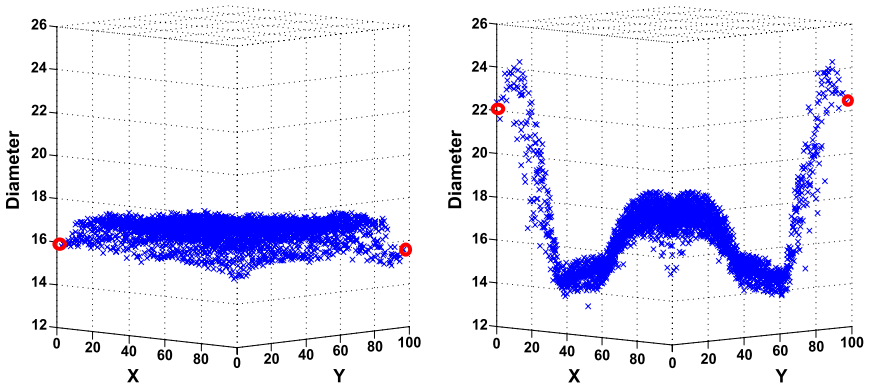


Figure 3. Local map of the circled node in Figure 2.

map is larger than the physical map. This is seen in the local map in Figure 3 where $2d = 49$ m.



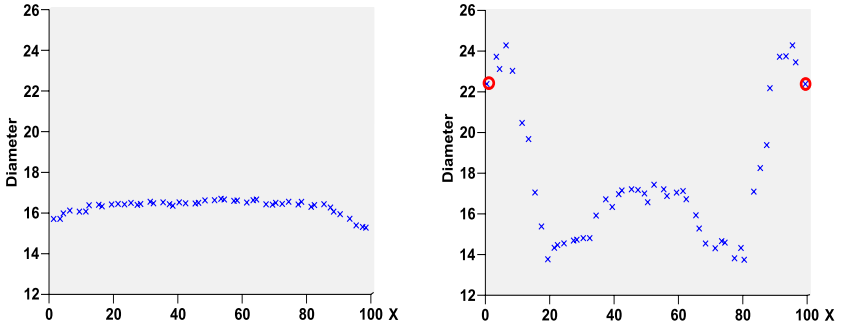
(a) Network without a wormhole (b) Network with a wormhole

Figure 4. Diameter measurements in a 2,500-node network.

To verify the effectiveness of the diameter feature in detecting wormholes, we computed the diameter for each node in the original 2,500-node network (Figure 2(a)) without and with a wormhole. The results are shown in Figures 4(a) and 4(b), respectively.

The diameters of the local maps of nodes close to a wormhole (i.e., near the circles in Figure 4(b)) are noticeably increased because of their proximity to the wormhole in comparison with the diameters for the same nodes in the network without a wormhole (Figure 4(a)). In Figure 4(b), the diameters of the local maps are roughly equal to R (14 to 18 m for $R = 15$ m) unless there is a wormhole attack, in which case the diameters of the local map become larger when the corresponding nodes are closer to the wormhole. On the other hand, the diameters of the local maps of nodes farther away from the wormhole or located in a distant part of the network (e.g., middle area in Figure 4(b)) are almost the same as those for nodes located in the same regions in Figure 4(a), which does not have a wormhole.

The diameter of a local map has the highest value (25 m) for nodes located about 7 m from the ends of the wormhole. The diameter values decrease for nodes closer to the network edges, but the values remain above 22 m.



(a) Network without a wormhole. (b) Network with a wormhole.

Figure 5. Diameter measurements in a 50-node network with a string topology.

The "diameter" feature is also effective at detecting wormholes in networks with irregular shapes and in networks with multiple wormholes. This was verified by conducting experiments on a network with a string topology and a network with two wormholes.

The string topology experiment involved testing a 50-node network whose nodes are uniformly distributed in a 100 m string in one dimension. First, the diameter was computed for each node in the network without any wormholes; as shown in Figure 5(a), the diameter is no more than 16.8 m. Next, a wormhole was added to the network; the ends of the wormhole were located at the two ends of the string. As shown in Figure 5(b), the diameters for nodes close to the wormhole ends are larger than 22 m.

To test the effectiveness of the "diameter" feature in detecting multiple wormholes in a network, we deployed two wormholes in the network of Figure 2(a). The diameter measurements for all the nodes are shown in Figure 6 (the shading bar indicates the diameter value). The locations of the ends of the two wormholes are represented as circles; the dashed lines are the wormhole tunnels.

Figure 6 shows that even when two wormholes are very close to each other, the peak diameter values still occur for nodes that are close to a wormhole end. The four peak values are 24.8 m, 25.2 m, 22.2 m and 22.6 m. Therefore, by computing the diameter d for a local map, the detection algorithm can run independently for each node and in conjunction with the computation of its local map. Since all the nodes in this area are within one/k hops of the calculating node, the detection algorithm can compute the diameters for the local maps after determining the location of each neighbor node.

Wormhole Detection Procedure The wormhole detection procedure is shown in Algorithm 2. The "diameter" feature is used to determine whether or not there is a wormhole attack. The experimental results in Figures 4(a)

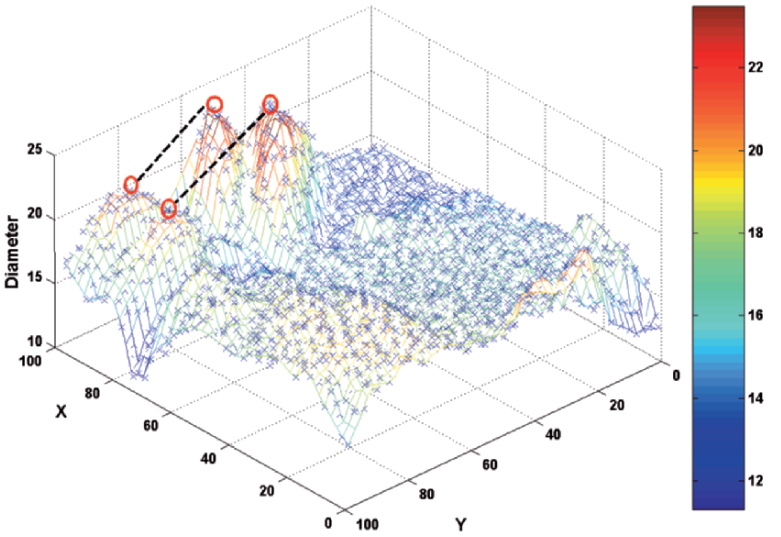


Figure 6. Diameter measurements in a 2,500-node network with two wormholes.

Algorithm 2 : Wormhole Detection Procedure (for Node a).

```
1: INPUT: local map G in node a for N_a ∪ {a}
2: diameter d = 0
3: for each b ∈ N_a ∪ {a} do
4:    for each node c ∈ N_a ∪ {a} − {b} do
5:       if 2d < distance(b, c) in local map G then
6:          2d = distance(a, b) in local map G
7:       end if
8:    end for
9: end for
10: if d > (1 + λ) × 1.4R then
11:    return "FOUND WORMHOLE" to sink node.
12: end if
```

and 4(b) show that the diameters for the local maps are around R when there is no wormhole. However, when there is a wormhole, the diameters for the local maps computed for nodes close to a wormhole end are higher (more than $1.5R$ in the example). Therefore, we can define a diameter threshold for detecting wormholes. Based on our experimental results, we define the threshold as $1.4R$ ($= 21$ m since $R = 15$ m). In general, the lower the value of the threshold, the higher the likelihood of false positives.

We introduce a parameter λ to adjust the sensitivity of the detection procedure. Suppose the diameter of a local map is d. Then, if $d > (1 + \lambda)1.4R$ (where λ is a constant between 0 and 1), there is a wormhole in the network. If there is no wormhole, the erroneous result is probably due to a localization error.

The detection step involves a computational cost of $O(|N_a|^2 n)$ and a memory cost of $O(|N_a|)$ per node. No communication cost is associated with this step.

5. Simulation Results

This section describes the simulation environment and presents the results of our simulation experiments.

5.1 Simulation Environment

The detection algorithm was implemented as a routing agent using ns-2 version 2.29 [10] with 802.15.4 MAC layer [19] and CMU wireless extensions [14]. The following configuration was used for ns-2: RF range = 15 m, propagation = TwoRayGround and antenna = Omni Antenna. The wormhole was implemented as a wired connection with much less latency than the wireless connections.

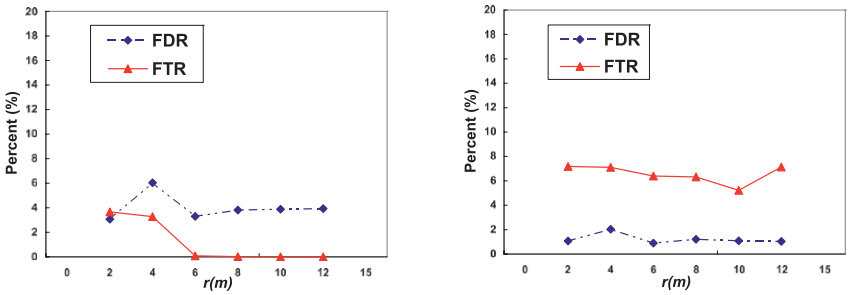
Uniform placement of nodes was used in the simulation experiments: n nodes were placed on a grid with $\pm 0.5r$ randomized placement error (r is the width of a grid square). We constructed a total of 24 placements for values of $n =$ 400, 900, 1,600 and 2,500, and $r = 2$, 4, 6, 8, 10 and 12 m. The reason for using uniform placement with $\pm 0.54r$ error is that it usually produces node holes and islands in just one placement. The location of the wormhole was completely randomized within the network.

5.2 Detection Results

As the value of λ is decreased, the accuracy of detecting wormhole attacks is increased, but the likelihood of false alarms is increased. To evaluate the accuracy of attack detection under different λ values, we introduce the following measures:

- **False Detection Rate (FDR):** This is the frequency with which a detection system falsely recognizes identical characteristics as being different, thus failing to tolerate, for example, a normal localization error. FDR is computed as the number of normal localization errors flagged as detected wormholes divided by the total number of trials.

  To compute an FDR value, we count the number of nodes that sent "FOUND WORMHOLE" messages but that are "far away" from the ends of a wormhole multiplied by the number of normal localization errors flagged as detected wormholes. We assume that if a node is $R = 15$ m away from the ends of a wormhole, then the node is essentially unaffected by the wormhole and is, therefore, considered to be "far away" from the



(a) $\lambda = 0$. (b) $\lambda = 0.1$.

Figure 7. FDR and FTR for various node spacings.

wormhole. An FDR value of zero means that there are no false alarms when detecting wormholes.

- **False Toleration Rate (FTR):** This is the frequency with which a detection system falsely recognizes different characteristics as identical, thus failing to detect a wormhole attack. FTR is computed as the number of wormhole attacks that are not detected divided by the total number of trials.

  If a wormhole is present in an experiment, but there is no node to send "FOUND WORMHOLE" messages, we count it as an undetected wormhole. Therefore, an FTR value of zero means that the detection algorithm is successful at detecting wormholes in all experiments.

We used the experimental setup described above with one wormhole in each placement. The FDR and FTR values for the experiments are presented in Figure 7. The detection algorithm has a low FTR value with FDR = 0 when $\lambda = 0$ as shown in Figure 7(a). When $\lambda = 0.1$, as shown in Figure 7(b), the detection algorithm achieves a low FDR value with FTR = 0.

6. Conclusions

The wormhole geographic distributed detection (WGDD) algorithm presented in this paper employs a hop counting technique as a probe procedure for wormholes, reconstructs local maps using multidimensional scaling at each node, and uses a novel "diameter" feature to detect distortions produced by wormholes. Unlike other wormhole detection algorithms, it does not require anchor nodes, additional hardware (e.g., directional antennas and accurate clocks) or the manual setup of networks. Even so, it can rapidly provide the locations of wormholes, which is useful for implementing countermeasures. Because the algorithm is distributed, each node can potentially detect the distortions produced by a wormhole, which increases the likelihood of wormhole detection.

Simulation results demonstrate that the algorithm achieves an overall detection rate of nearly 100% (with an FTR near zero as shown in Figure 7(a)). Even in case of shorter wormholes that are less than three hops long, the algorithm has a detection rate of over 80% (with an FTR of less than 20%). Furthermore, the algorithm can be adjusted to produce extremely low false alarm rates (with an FDR of zero as shown in Figure 7(b)).

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] S. Čapkun, L. Buttyán and J. Hubaux, SECTOR: Secure tracking of node encounters in multi-hop wireless networks, *Proceedings of the First ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 21–32, 2003.

[3] W. Du, L. Fang and P. Ning, LAD: Localization anomaly detection for wireless sensor networks, *Journal of Parallel and Distributed Computing*, vol. 66(7), pp. 874–886, 2006.

[4] L. Hu and D. Evans, Using directional antennas to prevent wormhole attacks, *Proceedings of the Eleventh Network and Distributed System Security Symposium*, pp. 131–141, 2004.

[5] Y. Hu, A. Perrig and D. Johnson, Wormhole Detection in Wireless Ad Hoc Networks, Technical Report TR01-384, Department of Computer Science, Rice University, Houston, Texas, 2002.

[6] Y. Hu, A. Perrig and D. Johnson, Packet leashes: A defense against wormhole attacks in wireless networks, *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1976–1986, 2003.

[7] J. Kong, Z. Ji, W. Wang, M. Gerla, R. Bagrodia and B. Bhargava, Low-cost attacks against packet delivery, localization and time synchronization services in underwater sensor networks, *Proceedings of the Fourth ACM Workshop on Wireless Security*, pp. 87–96, 2005.

[8] L. Lazos and R. Poovendran, SeRLoc: Robust localization for wireless sensor networks, *ACM Transactions on Sensor Networks*, vol. 1(1), pp. 73–100, 2005.

[9] D. Liu, P. Ning and W. Du, Attack-resistant location estimation in sensor networks, *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks*, pp. 99–106, 2005.

[10] S. McCanne and S. Floyd, The network simulator – ns-2 (nsnam.isi.edu/nsnam/index.php/User_Information), 2007.

[11] J. Newsome, E. Shi, D. Song and A. Perrig, The sybil attack in sensor networks: Analysis and defenses, *Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, pp. 259–268, 2004.

[12] P. Papadimitratos and Z. Haas, Secure routing for mobile ad hoc networks, *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2002.

[13] R. Poovendran and L. Lazos, A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks, *Wireless Networks*, vol. 13(1), pp. 27–59, 2007.

[14] The Rice Monarch Project, Wireless and mobility extensions to ns-2 (www.monarch.cs.cmu.edu/cmu-ns.html), 2007.

[15] M. Vieira, C. Coelho Jr., D. da Silva Jr. and J. da Mata, Survey of wireless sensor network devices, *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation*, vol. 1, pp. 537–544, 2003.

[16] W. Wang and B. Bhargava, Visualization of wormholes in sensor networks, *Proceedings of the ACM Workshop on Wireless Security*, pp. 51–60, 2004.

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

[18] Y. Xu, J. Ford and F. Makedon, A variation on hop counting for geographic routing, *Proceedings of the Third IEEE Workshop on Embedded Networked Sensors*, 2006.

[19] J. Zheng, Low rate wireless personal area networks: ns-2 simulator for 802.15.4 (release v1.1) (ees2cy.engr.ccny.cuny.edu/zheng/pub), 2007.