International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:7, 2009

Detecting and Locating Wormhole Attacks in Wireless Sensor Networks Using Beacon Nodes

He Ronghui, Ma Guoqing, Wang Chunlei, and Fang Lan

***Abstract*—**This paper focuses on wormhole attacks detection in wireless sensor networks. The wormhole attack is particularly challenging to deal with since the adversary does not need to compromise any nodes and can use laptops or other wireless devices to send the packets on a low latency channel. This paper introduces an easy and effective method to detect and locate the wormholes: Since beacon nodes are assumed to know their coordinates, the straight line distance between each pair of them can be calculated and then compared with the corresponding hop distance, which in this paper equals hop counts × node's transmission range R. Dramatic difference may emerge because of an existing wormhole. Our detection mechanism is based on this. The approximate location of the wormhole can also be derived in further steps based on this information. To the best of our knowledge, our method is much easier than other wormhole detecting schemes which also use beacon nodes, and to those have special requirements on each nodes (e.g., GPS receivers or tightly synchronized clocks or directional antennas), ours is more economical. Simulation results show that the algorithm is successful in detecting and locating wormholes when the density of beacon nodes reaches 0.008 per m^2.

***Keywords*—**Beacon node, Wireless sensor network, Wormhole attack.

I. Introduction

WIRELESS sensor networks (WSNs) are comprised of many small and resource constrained sensor nodes that are deployed in an environment for many applications thich require unattended, long-term operations. In WSNs, each node serves as a router for other nodes which allows data to travel by utilizing multi-hop network paths without relying on wired infrastructure. So, for those applications run in untrusted environments, such as emergency rescue and military operations, security issues is a major concern.

Several types of malicious attacks have been well described in the literature, they are generally categorized as mote-class attacks and laptop-class attacks, insider attacks and outsider attacks, passive attacks and active attacks [1],[3]. This paper focuses on the wormhole attacks, which belong to laptop-class, outsider, passive attacks. In a typical wormhole attack, an adversary tunnels messages received in one part of the network over a low-latency link(the wormhole link) and replays them in a different part [1],[2],[3],[7],[8]. The wormhole attack can affect secure location[10], network routing, data aggregation and clustering protocols[14]. This attack is particularly challenging to deal with since the adversary does not need to compromise any legitimate nodes or have access to any cryptographic keys[14].

This paper introduces an easy and effective method to detect and locate wormholes. The basic idea is to take advantage of the known locations of beacon nodes which originally are used in location discovery in WSNs. Note that it is the wormhole detection rather than the development of a new secure localization scheme that this paper focuses on. A hop counting technique is employed to make every beacon node know its hop distance to the other beacon nodes as well as the coordinates of them. Since some hop distances may be remarkably decreased by a wormhole link, while the corresponding straight line distances calculated by the coordinates are unaffected, following the law of mathematics (i.e. straight line distance is the shortest one between two points), when a straight line distance is larger than a hop distance by a threshold value, conclusions can be made that there exists wormhole attacks. Our algorithm can also provide an approximate location of a wormhole, which can assist in implementing defense mechanisms. Another advantage of our algorithm is that it can deal with multiple wormholes.

The rest of this paper is organized as follows. The next section reviews related work. Section III describes our wormhole detection algorithm and a method to locate the origin point and the end point of the wormhole link. Section IV analyses overheads and the localization precision of the algorithm. Section V presents our simulation evaluation on the proposed techniques, and section VI concludes this paper.

II. Related Work

A number of techniques have been proposed in recent years to detect wormhole attacks in WSNs. The solutions proposed attempt to bound the distance that any message can travel [2] or securely discover the set of one-hop neighbors [9],[13].

Packet Leashes[2] employ the notions of geographical and temporal leashes. Geographical leash insures that the recipient of the packet is within a certain distance from the sender. Temporal leash ensures that the packet has an upper bound of its lifetime (restricts the maximum travel distance). The assumption is that each sensor 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

He Ronghui, Ma Guoqing, Wang Chunlei, Fang Lan are with the Department of Network Technique, Beijing Institute of System Engeering, ChaoYang District, Beijing, China 100101 (e-mail: ronghuih@ gmail.com, gleader@ gmail.com, clwang01@gmail.com, flan@gmail.com).

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. Fine grain timing analysis is also used in [4].

In [5] a graphtheoretic framework is used to prevent wormhole attacks. The protocol uses "guard nodes" that know their "correct" locations, which is similar to beacon nodes used in this paper, but it assumes the guard nodes have higher transmit power and different antenna characteristics, which are unnessary in our approach.

LiteWorp [12] relies on overhearing by selected nodes, called Guards, distributed throughout the network. The guards monitor local control traffic to detect wormhole attacks. LiteWorp assumes overhearing, omnidirectional antennas, and a static topology, making it infeasible for large classes of networks.

Hu and Evans [9] detect wormholes by equipping directional antennas to each network nodes so they can all have the same orientation. Same equipments are used in SeRLoc [10], a distributed secure localization scheme. Wenliang Du, et al. propose an anomaly detection scheme named LAD [11] which needs the help of deployment knowledge to find out whether the estimated location is consistent with its observations.

III. Detecting and Locating Wormhole Attacks

Since a wormhole attack is passive, it occurs only when a message is being transmitted in the region near a wormhole. To detect and locate a wormhole attack, A distributed algorithm is used, in which each beacon node acts as a detector, each sensor node participates in hop counting, while the base station controls the start and end of the detecting process, and estimate the locations of wormhole ends based on alarm messages sent from beacon nodes.

A. Network Assumptions and Thread Model

First, assume that the network consists of a base station, a set of sensor nodes *S* of unknown location and a set of beacon nodes *B* which already know their absolute locations via GPS or manual configuration. Assume that all network nodes are deployed randomly in a specific network region of area *A*.

Assume that the beacon-to-sensor communication range is the same with the sensor-to-sensor communication range. It is also assumed that the communication channels are bidirectional, i.e. if a node *a* can receive a message from *b*, then *b* can also receive a message from *a*.

Assume that all beacon nodes are uniquely identified. In other words, a node can identify the original sender of each beacon packet based on the cryptographic key used to authenticate the packet. This can be easily achieved with a pairwise key establishment scheme [15],[16] or a broadcast authentication scheme [17].

Assume that a wormhole link is bidirectional with two endpoints(wormhole nodes), and network nodes are not compromised by attackers.

B. Wormhole Detection Algorithm

Before describing the detecting algorithm, we'd like to explain two terms used in this paper first.

Hop count the hop count between two beacon nodes means the minimum number of hop-by-hop transmissions to reach one beacon node from another.

Hop distance the hop distance between two beacon nodes refers to the sum of each hop size in meter.

In many localization schemes, average hop size is used to estimate the hop distance between an arbitrary node and a beacon node. For example, DV-hop [6],[18] computes average hop size as:

$$hopsize_i = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i} h_j}, \quad i \neq j$$

Where (x_i, y_i), (x_j, y_j) are the coordinates of beacon nodes *i* and *j*, and h_j is the hop count between them.

In this paper, since estimating each sensor node's location as precisely as possible is not our purpose, it is reasonable to assume that the communication range *R* of each node in the WSN is the same, and adopt it as the average hop size. Hence the hop distance becomes $hopcount \times R$. Obviously, this assumption introduces inaccuracy to the hop distance by overestimating the hop size. But it will be showed later that it does not tamper with the wormhole detection. On the contrary, it makes the algorithm simple, which is important for a resource restricted system as WSNs.

In a benign wireless sensor network, hop distance between any pair of beacon nodes derived from above is certainly larger than the distance calculated by their locations. But at the presence of a wormhole, things become different especially when the beacon nodes distributed near the wormhole ends are concerned. The hop count between them may be reduced on a large scale, so as to the hop distance is much less than the distance based on their locations. This is called as an ***abnormity***.

Our detection algorithm is able to discover such abnormities and alarm the existence of wormhole attacks. To order to achieve this, two kinds of messages are defined, probe message and alarm message.

A probe message is the message broadcasted by each beacon node at the beginning of the detecting process to probe the hop counts between itself and other beacon nodes. It contains a probe vector like

ID_i	$Coordinator(ID_i)=(x_i, y_i)$	*hopcount*

Where ID_i is the unique identification number of beacon node B_i, (x_i, y_i) is its location, *hopcount* is a counter initialized with zero, and will gain an increment each time the message is forwarded by a sensor node. When a probe message sent by B_i arrives at another beacon node B_j, the *hopcount* field records the hop count between them. The receiver then calculates their hop distance and straight line distance using information extracted from the probe message and makes his judgment. An alarm message is sent immediately to the base station in case of an abnormity, containing information like

ID_j	ID_i	*hopcount*

In which ID_j indicates the alarm sender B_j, and ID_i indicates the beacon node from whose probe message the abnormity is detected, i.e. B_i. *hopcount* is used to help the base station decide which beacon pair in received alarm messages is the nearest one to wormhole ends. The next section describes this problem in detail.

To make sure that a node (sensor or beacon) only disposes the first comer of probe messages originated from a specific beacon node (for the first comer is thought to be the one that comes through the shortest path), a set *Q* is maintained by each node to record the *IDs* of beacon nodes whose probe message has been forwarded by the node.

In our distributed detection algorithm, each beacon node acts as a detector, and each sensor node participates in hop counting. Algorithm 1 illustrates the detect procedure for each sensor node. The detect procedure for each beacon node is presented in algorithm 2.

Algorithm1 Detect procedure for each sensor node

Initialize: Q = null
for each probe message received and not (TIMEOUT or WORMHOLE DETECTED) **do**
  extract *id* and *hopcount* from probe message
  if $id \in Q$ **then**
    drop (*probe message*)
  else
    $Q = Q+\{id\}$
    hopcount = *hopcount* +1
    Forward (*probe message*) to MAC
  end if
end for

Algorithm2 Detect procedure for each beacon node B_i

Broadcast its own probe message
for each probe message received and not (TIMEOUT or WORMHOLE DETECTED) **do**
  extract *id*, *hopcount* and (x_j , y_j) from probe message
  if $id \in Q$ **then**
    drop (*probe message*)
  else
    $Q = Q+\{id\}$
    hopcount = *hopcount* +1
    if $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - hopcount \times R > 0$ **then**
      send alarm message to base station.
    else
      Forward (*probe message*) to MAC
    end if
  end if
end for

C. Wormhole Localization Algorithm

In a well synchronized network, in which all the beacon nodes can start broadcasting their probe messages at the same time, it is reasonable to assume that the first alarmed node is the nearest one to a wormhole end. If that is the case, the base station can stop the detecting procedure and take the coordinates in the first alarm message as the approximate locations of wormhole ends. But considering that the difference of transmission distances from alarmed beacon nodes to the base station may influence the arriving sequence of alarm messages, and the situation of multiple wormholes, it is not enough for the base station to draw a conclusion based on the first arrived alarm message only.

Fig. 1 illustrates a circumstance of two wormholes in a cross configuration. Beacon nodes A,B,C,D,E,F,G,H are placed near to wormholes, in which A,B,E,F are one-hop away from a certain wormhole end. According to our detecting process, if it does not get timeout, early or late, the base station will receive alarm messages from them with information like: *<A,B,h1>*, *<B,A,h1>*, *<A,D,h2>*, *<D,A,h2>*, *<C,D,h3>*, *<D,C,h3>*, *<C,B,h4>*, *<B,C,h4>*, *<E,F,h5>*, *<F,E,h5>*, *<E,H,h6>*, *<H,E,h6>*, *<G,H,h7>*, *<H,G,h7>*, *<G,F,h8 >*, *<F,G,h8 >*.

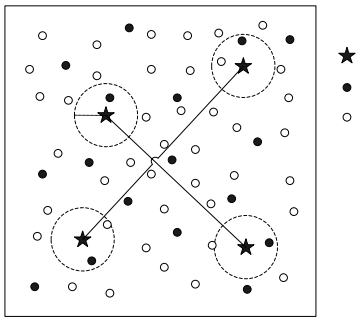

Fig. 1 Two wormholes in a cross configuration

Obvious redundancies exist in these messages. To find out the two wormholes and locate them as accurately as possible, two things must be settled. First, which of these alarms are arose from one wormhole, in other words, how many wormholes are there in the network. Second, for those alarmed beacon pairs, which is the nearest one to wormhole ends.

Algorithm 3 shows the localization algorithm of the base station, in which two operations are introduced: ***duplicate*** and ***nearto***. The algorithm takes *<A,B,h1>* as a ***duplicate*** of *<B,A,h1>*, hence drop the later came one. ***Nearto*** is used to judge if two beacon pairs are near to each other: by saying beacon pair(*A*,*B*) ***nearto*** beacon pair(*C*,*D*), it means the distance between node *A*(*B*) to node *C*(*D*) is smaller than a threshold. For those beacon pairs ***nearto*** each other, the one who has minimum *hopcount* value is saved and others droped. A_MSG is a set maintained by the base station to store alarm messages which can be used to most accurately locate wormholes. This algorithm keeps only one message for each wormhole.

In the above example, after localization process, two alarm messages are left in A_MSG: *<A,B,h1>* and *<E,F,h5>*, which means there exists two wormhole links, and beacon pair (*A*,*B*) and (*E*,*F*) are the approximate locations of wormhole ends. It can seen from fig. 1 that the localization error is less than 1*R*. That means, take *A* for example, a wormhole end is located

within a circle of radius *R* centered at *A*.

Algorithm3 localization procedure for the base station

```
Initialize: A_MSG = null
INPUT alarm message <ID_i, ID_j, hopcount >
start: for each alarm message <ID_i, ID_j, hopcount > and not
   TIMEOUT do
   if A_MSG = null then
      A_MSG = {< ID_i, ID_j, hopcount >}
   else
      for each a_msg in A_MSG do
         if <ID_i, ID_j, hopcount > duplicate a_msg then
            drop (alarm message)
            goto start
         end if
         if (ID_i, ID_j) nearto (a_msg.ID_i, a_msg.ID_j) then
            if hopcount < a_msg. hopcount then
               a_msg = <ID_i, ID_j, hopcount >
            else
               drop(alarm message)
            end if
            goto start
         end if
      end for
      A_MSG = A_MSG +{<ID_i, ID_j, hopcount >}
   end if
end for
return A_MSG
```

IV. Analysis and Improvements

A. Overheads

This section analyzes the overheads of our wormhole detection algorithm.

Let *B* denotes the set of beacon nodes, *S* denotes the set of sensor nodes, and |.| denotes the cardinality of a set. The random deployment of the network nodes in an area *A* can be modeled as a spatial homogeneous Poisson point process[10]. Let N_a be the set of one-hop neighbors of a node *a*. the probability that *a* has *k* neighbors $P(|N_a| = k)$ is equal to the probability that *k* nodes are deployed within an area of size πR^2, where *R* is the transmission range:

$$P(|N_a| = k) = \frac{(\rho\pi R^2)^k}{k!} e^{-\rho\pi R^2} \quad (1)$$

In which $\rho = \frac{|B|+|S|}{A}$ is the density of the nodes.

Our algorithm has a memory cost of O(|*B*|) per node and a computational cost of O(|N_a ||*B*|) per node in the worst condition, that is, there is no wormhole in the network.

The communication cost is relatively high because of the flood-based approach for the hop counting procedure. But the existence of the wormholes will reduce the communication cost on a large scale, for the wormholes make two distant nodes look like instant neighbors, alarms will soon arise after several hops transmission and the detection procedure can be terminated long before the flood procedure is over.

According to the information redundancy, several improvements can be made to reduce the cost of our algorithm further.

1) It is unnecessary for each beacon node of a beacon pair that detected an abnormity to send an alarm message. Making the one nearer to the base station rise the alarm and the other one keep silent is enough.
2) It is effective to have all the beacon nodes in the network began to broadcast simultaneously, whereas the communication cost is high. An iterative method can be used instead. Firstly, two or more beacon nodes (the distance of them must larger than a threshold) are selected as the detectors. Then, based on their detection results, some more beacon nodes will be added in and the detection procedure runs once again. The iteration will not stop until acceptable localization accuracy is achieved or it gets timeout.

B. Precision of Localization

This section investigates the impact of the beacon node density in the precision of localization.

As illustrated in Fig. 2, if beacon nodes *A* and *B* are selected as the estimated locations of two ends of a wormhole, then the localization error *LE* is calculated as $LE = \frac{d_1 + d_2}{2R}$.

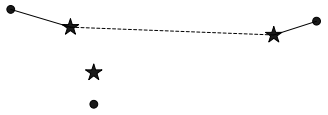

Fig. 2 Localization error illustration

To achieve a localization error less than 1, at least one beacon node must be located within the circle of radius *R* centered at each wormhole end. Let *W* be the set of wormhole nodes, N_w be the set of one-hop neighbors of a wormhole node *w*, the probability that every wormhole node has at least one beacon neighbor can be computed using (1):

$$P(|N_w| \geq 1, \forall w \in W) = P(|N_w| \geq 1)^{|W|}$$
$$= (1 - P(|N_w| < 1))^{|W|}$$
$$= \left(1 - e^{-\rho_B \pi R^2}\right)^{|W|}$$

Where $\rho_B = \frac{|B|}{A}$ is the density of beacon nodes.

Fig. 3 shows the probability of *LE*<1 for different ρ_B with one wormhole(|*W*|=2)and two wormholes (|*W*|=4).

It can seen from fig. 3 that to achieve a *P*(*LE*<1) = 90%, a beacon node density of 0.01 per m^2 is needed.

V. Performance Evaluation

A. Simulation Setup

250 sensor nodes were randomly distributed within a 50m*50m rectangular area. The beacon nodes were also randomly placed within the same area. The transmission range *R* of both sensors and beacon nodes was set as 10. The location

of the wormhole was fixed in the network.

To evaluate the accuracy of attack detection and localization under different ρ_B, sensors was distributed in advance , and ρ_B was changed from 0.002 to 0.02 step with 0.002. Considering the randomization of nodes placement may influence the detection results, this experiment was repeated 20 times.

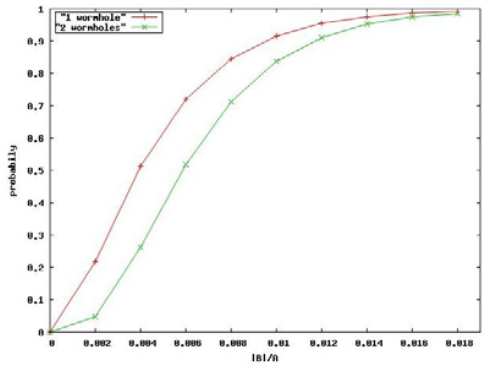

Fig. 3 $P(LE<1)$ for different ρ_B with 1 wormhole and 2 wormholes

B. Detection Results

Let's introduce false toleration rate(FTR) as the frequency with which a detection procedure fails to detect a wormhole attack. FTR is computed as the number of wormhole attacks that are not detected divided by the total number of attacks.

Another rate is calculated to estimate the localization precision, that is, the rate of $LE<1$. It is computed as the number of wormholes which $LE<1$ divided by the total number of attacks.

Fig. 4 shows the FTR and the $LE<1$ rate for the experiments. The detection algorithm has a high LE<1 rate with FTR=0 when $\rho_B \geq 0.008$, which means the algorithm is successful at detecting and locating wormholes in all experiments when $\rho_B \geq 0.008$. No false alarms are appeared in our experiments, that means the value of false positive is zero.

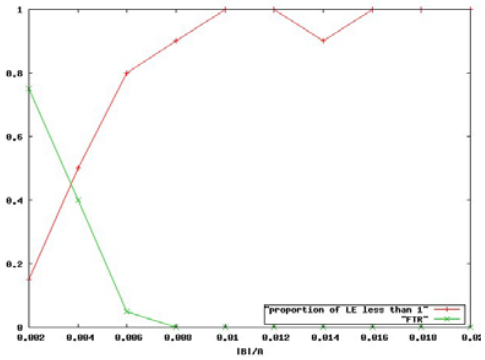

Fig. 4 FTR and LE<1 rate in our experiments

VI. Conclusion

This paper presents a distributed wormhole detection and localization algorithm which takes advantage of the known locations of beacon nodes. Its calculation cost is very low compareing to those require additional hardwares(e.g., directional antennas and accurate clocks) or manual setup of networks. Further more, it can provide the locations of wormholes with a localization error less than 1R when the density of beacon nodes reaches a certain small value.

References

[1] S. Shankar Sastry, Tanya Gazelle Roosta, "attacks and defenses of ubiquitous sensor networks," pp.10-22, Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-58.html

[2] Y.C. Hu, A. Perrig, and D.B. Johnson. Packet leashes, "A defense against wormhole attacks in wireless ad hoc networks," in *Proceedings of INFOCOM 2003*, April 2003.

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

[4] J. Eriksson, S. Krishnamurthy, and M. Faloutsos, "Truelink: A practical countermeasure to the wormhole attack," in *ICNP*, 2006.

[5] R. Poovendran and L. Lazos, "A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks," *ACM Journal of Wireless Networks (WINET)*, 2005.

[6] D. Nicolescu and B. Nath, "Ad-Hoc Positioning Systems (APS)," in *Proc. of IEEE GLOBECOM 2001*, San Antonio, TX, USA, November 2001.

[7] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," *Proc. SCS Communication Networks and Distributed Systems Modeling and Simulation*, 2002.

[8] K. Sanzgiri, B. Dahill, B. Levine, C Shields, and E. Belding-Royer, "A secure routing protocol for ad hoc networks," in *ICNP*, 2002, pp. 78– 89.

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

[10] L. Lazos, and R. Poovendran, "SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks," *Proceedings ACM WiSe* (October 2004).

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

[12] I. Khalil, S. Bagchi, and N. B. Shroff, "LITEWORP: A Lightweight Countermeasure for the Wormhole attack in multihop wireless network," in *International Conference on Dependable Systems and Networks* (DSN), 2005.

[13] S. ˇ Capkun, L. Buttyan, J. Hubaux, "SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks," *Proceedings of Security of Ad Hoc and Sensor Networks* (Oct. 2003) pp. 21~32.

[14] Khin Sandar Win, Pathein Gyi, "Analysis of Detecting Wormhole Attack in Wireless Networks," *Proceedings Of World Academy Of Science Engineering And Technology* Volume 36 December 2008.

[15] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *IEEE Symposium on Research in Security and Privacy*, 2003, pp. 197–213.

[16] W. Du, J. Deng, Y. S. Han, and P. Varshney, "A pairwise key predistribution scheme for wireless sensor networks," in *Proceedings of 10th ACM Conference on Computer and Communications Security* (CCS'03), October 2003, pp. 42~51.

[17] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar, "SPINS: Security protocols for sensor networks," in *Proceedings of Seventh Annual International Conference on Mobile Computing and Networks*, July 2001.

[18] Niculescu D, Nath B, "DV based positioning in ad hoc networks[J]," *Journal of Telecommunication System* 22(1/4), 2003, pp.267~280.