2009 15th International Conference on Parallel and Distributed Systems

WormCircle: Connectivity-based Wormhole Detection in Wireless Ad Hoc and Sensor Networks

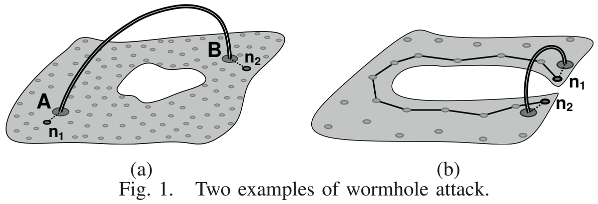
Dezun Dong[*][†], Mo Li[†], Yunhao Liu[†], and Xiangke Liao[*]
[*]School of Computer, National University of Defense Technology, Changsha, China
[†]Dept. of Computer Science and Engineering, Hong Kong University of Science and Technology

***Abstract*—Wormhole attack is a severe threat against wireless ad hoc and sensor networks. It can be launched without compromising any legitimate node or cryptographic mechanisms, and often serves as a stepping stone for many serious attacks. Most existing countermeasures often make critical assumptions or require specialized hardware devices in the network. Those assumptions and requirements limit the applicability of previous approaches. In this work, we explore the impact of wormhole attacks on network connectivity topologies, and develop a simple distributed method to detect wormholes, called WormCircle. WormCircle relies solely on local connectivity information without any requirements on special hardware devices or making any rigorous assumptions on network properties. We establish the correctness of this design in continuous geometric domains and extend it into discrete networks. We evaluate the effectiveness in randomly deployed sensor networks through extensive simulations.**

I. Introduction

Wireless ad hoc and sensor networks are emerging as promising techniques for many important applications such as homeland security, military surveillance, environmental monitoring, and target detection and tracking etc. Many of these applications involve a large number of sensors distributed in a vast geographical area. Security is crucial for those mission-critical applications, which often work in unattended and even hostile environment. One of the most severe security threats in ad hoc and sensor networks is wormhole attack, which has been independently introduced in previous works [1] [2] [3].

Figure 1 (a) displays a classic example of a wormhole attack, in which an adversary initially establishes a high-speed out-of-band link between two points in a multihop wireless network. The attacker's link is referred to as a wormhole link or simply a *wormhole*. The two ends of a *wormhole link* are *wormhole endpoints*. In this example, AB represents a wormhole link in the network connecting two distant areas. The adversary can capture and replay the packet signals in the physical layer or simply retransmit the packet in the link layer [3]. In this case, as illustrated in Figure 1 (a), node n_1 and node n_2 can communicate directly as if they were direct neighbors. By establishing these wormhole tunnels, the attackers are able to attract and control a large amount of network traffic so as to launch a variety of attacks, such as, forward packets out of order, selectively drop specified packets, etc. More severely, by gathering packets, adversaries are able to analyze network traffic for cipher breaking, protocol reverse engineering, etc. Hence, wormhole attack can serve as a stepping stone for many other more aggressive and severe attacks, and significantly imperil routing, localization, topology control, as well as many other network protocols [3]. Wormhole attack can be mounted in a passive mode without modifying any packet. Thus, a wormhole attack can be launched successfully without compromising any legitimate node or breaking cryptographic mechanisms, and cannot be defended effectively by using only cryptographic techniques [3].

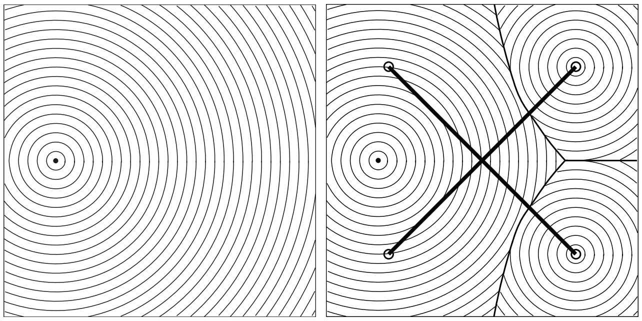


(a) (b)
Fig. 1. Two examples of wormhole attack.

The wormhole attack problem has received considerable attentions recently. Many countermeasures have been proposed to detect wormholes in wireless ad hoc and sensor networks. Some approaches [3] [4] explore node locations information to check the violence of communication range bound. Some methods utilize tight global clock synchronization [3] or special hardware equipped [5] etc. to verify the packet transmission latency due to wormholes relay. Some approaches capture the existence of physical infeasibility links by neighbor witness, which require the directionality of antenna communication [6], attack-free environment during the deployment phase [7], and the assistance of some location-aware mobile node [8]. Some approaches utilize communication graph constraints to detect wormholes. Such methods assumes the existence of guard nodes with extraordinary communication range [9], a central controller calculating the network layout [10], UDG graph model [11]. Other approaches use statistical analysis to catch abnormal routing selection [12], increased neighbor number, and decreased lengths of shortest paths [13] etc. To summarize, most existing solutions of wormhole detection require specialized hardware devices or making strong assumptions, which largely restrict their applicability in large-scale resource-constrained sensor networks. It is of great necessity and challenge to design an effective wormhole detection method while relaxing those critical assumptions.

In this design, we develop a simple distributed algorithm for wormhole detection in wireless ad hoc and sensor networks, using only the communication graph, and not making unrealistic assumptions. Our method does not assume any special hardware devices, special guard nodes, or location measurements (including angular or distance information). More importantly, we do not force that the communication

1521-9097/09 $26.00 © 2009 IEEE
DOI 10.1109/ICPADS.2009.97



(a) (b)

Fig. 2. WormCircle in continuous domains. (a) Wavefront circles with no wormholes; (b) Wavefront circles with two wormholes.

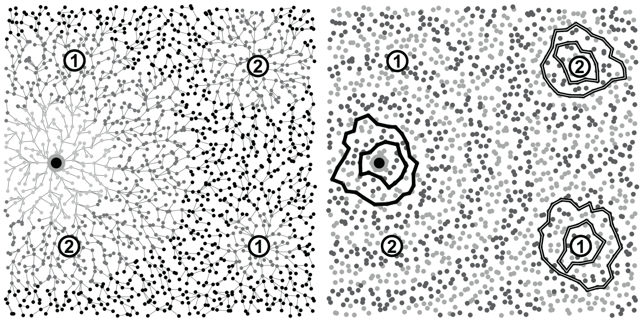
graph follows the unit disk graph model or quasi unit disk graph model. Specifically, our wormhole detection algorithm is motivated by an observation that a legitimate multihop wireless network deployed on the surface of a geometric terrain (possibly with irregular boundaries, inner obstacles), while the wormholes in the network inevitably change the network connectivity topology, resulting in some forbidden structures that we call wormhole circles. Our method locates the wormhole by identifying wormhole circles.

This paper is organized as follows. We first describe the basic principle of WormCircle in Section II and present the localized WormCircle protocol in Section III. In Section IV, we evaluate our design through comprehensive simulations, and conclude this work in Section V.

II. Basic WormCircle Algorithm

In this section, we present our initial attempt, WormCircle, which detects wormholes by observing variations on network topologies. Normally, a wireless multihop network is deployed on the surface of a geometric environment, such as a plane or a rough terrain. In the continuous setting, a legitimate network is considered as a geometric surface (plane, terrain) area with a certain number of boundaries (including inner holes). We refer to the surface of the legitimate network as the *original surface*. A wormhole link is a continuous line segment of short length that connects two points on the surface. We first explain the main idea of WormCircle in a continuous space, and then describe the detection algorithm in discrete networks. At the end of this section, we summarize the advantages and limitations of WormCircle.

In the paper, we adopt the typical assumptions on wormholes. Adversaries launch a wormhole attack at the physical layer, and do not need to hold any legitimate network identification, compromise any cryptographic resources or network nodes. We assume that the nodes deployed by the adversary have not valid network entity and do not become part of the network, which makes the wormhole attack unobservable to the upper layers of the network. We assume that existing cryptographic authentication mechanisms ensure the integrity and authenticity of the replayed messages in routing and transport layer, such as symmetric cryptography [14] [15]. Furthermore, the adversary can place nodes at arbitrary places in the network.



(a) (b)

Fig. 3. WormCircle approach in discrete networks. (a) Shortest path tree; (b) Isoline circles around wormhole endpoints.

A. WormCircle Principle

The idea of WormCircle comes from a physical phenomenon in the wave propagation. Let us consider a network deployed with high density on a Euclidean plane. We select a root point s and initiate a circular wave from s. On the continuous plane, the points with the same distance to s form a wavefront around s, and the propagation of the wave can be treated as the process of the wavefront growing from s. When there are no wormholes, the wavefront of distance t from s forms a t-distant isoline around s (The isoline might be broken by holes on the plane and we will discuss it later), as shown in Figure 2 (a). On the other hand, if there exist wormholes, an interesting phenomenon happens. As shown in Figure 2 (b), bold lines depict two wormholes in the network. When the wave front arrives at one endpoint of a wormhole, it rapidly goes through the wormhole link and at the other endpoint of the wormhole and the outgoing wavefront forms small circles around the wormhole endpoint. We borrow a physical term here and name such circles wormhole diffraction circles (*wormhole circles* for short). The wormhole circles are specific symptoms in the network infected by wormholes. The main idea of WormCircle is to detect the wormhole circles, and then locate the wormhole accurately.

To detect a wormhole circle, we need to differentiate it with the legitimate isoline circles around the root point. There is an obvious difference between them. For the legitimate isoline circle C in the plane, its perimeter is $P_C = 2\pi R$, where R is the isoline distance. For the wormhole circle C_w with the same isoline distance R, however, its perimeter $P_{C_w} = 2\pi(R - r)$ is much smaller than the expected length $2\pi R$, where r is the distance from the import endpoint of the wormhole to the root. Thus, we are able to distinguish the two types of circles by tracing their perimeters and distance from the root.

B. WormCircle Algorithm in Discrete

We validate the principle of WormCircle in the discrete wireless networks, and design a distributed algorithm to detect the wormhole circles. We apply the node connectivity to verify the local continuity and utilize network hop count to approximate the real distance. Similar techniques of distance approximation also have been used by Wang et al. in their boundary detection algorithm [16]. Specifically, WormCircle

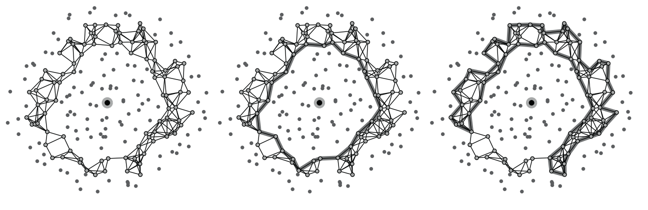


Fig. 4. Defining wormhole circles in the discrete. (a) The connectivity graph among isoline nodes, (b) A good isoline circle, (c) A defective circle.

first selects a root node and build a shortest path tree, and labels each node with a level according to its distance to the root node. WormCircle then traces the perimeter along the nodes of the same level and detects the isoline circles through their connectivity. We explain the procedure by an example shown in Figure 3, which is a discrete version of example in Figure 2. Endpoints of the two wormholes residing in this example are identified as 1 and 2. Figure 3 (b) illustrates the detection result. The isoline circles are detected and shown by single-line and double-line circles. The bold single-line circles around the root node are legitimate as their perimeters are compliant to their isoline distance level, but the double-line circles around the outgoing ends of the two wormholes are apparently of smaller perimeters to their isoline levels. We then present the details on how to launch WormCircle distributedly.

1) Labeling the Level of Node:

Initially, a root is randomly selected from the nodes in a distributed manner. The root floods the network and a shortest path tree is built. The shortest path tree is not flat due to the existence of wormholes, as shown in Figure 3 (a). Once the shortest path tree has been constructed, each node knows its minimum hop count to the root. The nodes with the same hop count fall in the same isoline, and are said to be of the same level. Apparently, the selection of root determines the level of each node and the structure of isolines. It thus affects the detection accuracy of WormCircle. For example, the symptom of wormhole circles will disappear if the selected root node is of equal distance to the two endpoints of a wormhole. We discuss such cases in the later.

2) Tracing Wormhole Circles:

In the discrete network, tracing the perimeter of circles is nontrivial. One major problem is how to define the counterpart of the continuous circle in the discrete network. Let us consider the example as shown in Figure 4 (a), which is a snapshot captured from a part of the entire network. The nodes, 4 or 5 hops away from the center big node, form a belt-shaped connected component which surrounds the center node. Our goal is to find a good 'discrete' circle in the belt component so that it truthfully reflects the skeleton of the belt (or strip) and surrounds the center node. The bold line cycle in Figure 4 (b) shows such an good candidate circle, while another cycle in Figure 4 (c) apparently does not properly reflect the skeleton of the component. How can we distinguish the two cases without location information? We later present the details about tracing wormhole circles. Our approach ensures to avoid the improper output as shown in Figure 4 (c). After the wormhole circles are identified, the endpoints of the wormhole are restricted within the circles. WormCircle derives the outbound endpoint of wormhole from tracing the wormhole circles along the levels and locating the origin of the wormhole circles. Thus the inbound end of the wormholes, adjacent to outgoing end, is also located.

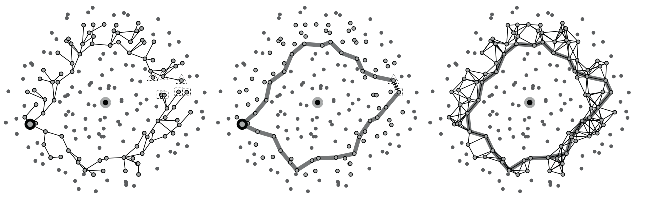


Fig. 5. Tracing wormhole circles. (a) Building a shortest path tree, (b) Finding a candidate circle, (c) Final isoline circle.

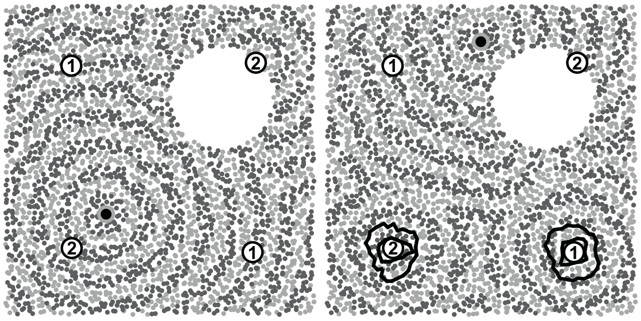
C. Tracing Wormhole Circles in Details

We here present the details about tracing wormhole circles. We explain the processes by tracing the isoline circles in the connected graph shown in the Figure 5. Suppose that the big dot in the center is the root and the circle nodes in Figure 5 show the strip of same level.

1) Finding Isoline Circles:

WormCircle first launches a restricted flooding from an arbitrary node in the strip and builds a shortest path tree, as shown in Figure 5 (a). The big circle is the selected root node and lines show the shortest path tree. In this shortest path tree, there are some pairs of nodes connected with each other but with their Least Common Ancestor (LCA) far away, shown as triangle and square nodes in Figure 5 (a). They are called as *cut pairs* [16]. WormCircle select one arbitrary cut pair among the leaf nodes and connects them, as shown in Figure 5 (b). The dot line connects two cut pair nodes, the triangle and square nodes. Then, WormCircle can trace back from the two cut pair nodes to obtain a candidate cycle, denoted by the grey lines in Figure 5 (b). Note that due to the insufficient density and randomness of node deployment, the nodes of the same level L sometimes may not be able to connect themselves together. WormCircle relaxes the constraints and traces the circles among a node strip with width k (e.g., for $k = 2$, the strip is composed of nodes with level L and $L + 1$). In the example in Figure 5, the isoline circle is in the strip of level 4 with width 2.

We then discuss more details about the cut pairs, and how we ensure to find a proper cycle to capture the outline of the strap trustworthily, and avoid the improper output as shown Figure 4 (b). Wang et al. originally define the cut pairs by two parameters δ_1 and δ_2 [16]. Informally, in our problem parameter δ_1 qualifies the length of each path that composes the isoline circle, while parameter δ_2 restricts that the two paths are not 'too close' to each other. In our isoline tracing, it is typically sufficient to avoid improper cases when we choose δ_2 to be a constant greater than the width of strip k (e.g., experiments reported in this paper use $\delta_2 = k + 1$). Thus, parameter δ_1 actually does not be utilized in our experiments,



(a) (b)

Fig. 6. Failure and improvement of basic WormCircle.

which is also due to that the length of candidate isoline circles in different level would vary greatly.

2) Contracting Rough Cycle:

The found candidate isoline circles still are rough, and maybe do not reflect the perimeter of a strip correctly. We further adjust a candidate isoline circle into a chordless cycle. Given a graph G, a cycle C in G is chordless if C is exactly the subgraph of G induced by all vertices in C. We refer to C as a chordless cycle of length k if C contains k vertices. The step of contracting a rough cycle is achieved in a distributed and iterative fashion. In each round, given the current candidate circle C, WormCircle sequentially labels each node with a number as its ID in circle C. These IDs are well-ordered along the circle. Each node in C, say node i, knows its direct neighbors N_i, and can locally tests whether existing node j in N_i and $j > i+1$. If so, node i reports a potential shrinkage by replacing its neighbor node $i+1$ with node j. After the shrinkage a new cycle is formed. If the circle cannot be further contracted, the iteration terminates and yields a chordless cycle. Note that in each iteration if multiple neighboring nodes claim shrinkage operations, they may conflict and only one operation is selected. Additive mechanisms can be used to break the symmetry, that is, node with lower ID has more high priority to shrink first. In the example, the bold lines in Figure 5 (c) show one obtained isoline circle after contracting operations.

3) Identifying Wormhole Circles:

After detecting the isoline circles, WormCircle estimates the perimeter of the circle and compares it with the isoline level to determine whether or not the detected isoline circle is a wormhole circle. As before mentioned in the continuous case, for the legitimate isoline circle C in the plane, its perimeter $P_C = 2\pi R$, where R is the isoline level. In the discrete network, we qualify a legitimate circle by the ratio between the perimeter and the level. The legitimate ratio is required to be greater than a threshold τ. (in most our experiments, $\tau = 5 < 2\pi$ is a proper selection).

D. Summary on WormCircle

In this subsection, we discuss the advantages, limitations and possible improvements for basic WormCircle. Compared with most existing approaches, WormCircle has many apparent advantages. First, since WormCircle relies solely on the network topology, it does not require any special hardware assistance, neither any rigorous assumptions on the network properties like UDG graph model, awareness of locations, Euclidean distance or time measures. Second, WormCircle is effective, since the symptom of wormhole circles is prevalent for general wormhole attacks. Using only connectivity makes WormCircle be able to reflect and capture inherent topological impacts introduced by wormholes. WormCircle potentially can detect some wormholes that cannot be detected by approaches based on Euclidean distance mismatch. For example, consider the network shown in Figure 1 (b). The Euclidean distance between node n_1 and n_2 could be little and even within the maximum possible communication range of the two nodes, but they simply cannot directly communicate due to the obstacle or disturbance between them. Hence, the current shortest communication path between node n_1 and n_2 in the network is a long journey, denoted as the black lines in Figure 1 (b). If an external link, as the double-line, is inserted into the network connecting n_1 and n_2, the two nodes then are able to communicate directly and the shortest path between them is shortened remarkably, which also significantly influences the communication between many other nodes. Such a wormhole attack clearly is hard to detect by approaches based on Euclidean distance mismatch, as the geometric distance does not correctly reflect the network communication path.

Although WormCircle offers the favorable applicability and demands on least assumptions, it is not perfect. In some cases, the symptom of wormhole circles may become inconspicuous or even invisible. As previously mentioned, when the two end points of a wormhole are of nearly equal hop counts to the root node, there will be no wormhole circles formed around the wormhole ends. Figure 6 (a) illustrate the case. There are no wormhole circles emerging around the endpoints of wormhole labeled as 1. When the outgoing end of the wormhole locates at the network boundaries (inner or outer), the wormhole circle is split by network boundaries and again cannot be detected by WormCircle. The wormhole labeled as 2 in Figure 6 (a) shows the case. The failure of WormCircle is mainly due to the unfavorable choice of the root node in the process of constructing the shortest path tree. For the same network shown in Figure 6 (a), when the root is moved to another position as shown in Figure 6 (b), WormCircle can detect both two wormholes correctly. It is clear that the detection rate can be improved if WormCircle is launched multiple times independently, when multiple different root nodes are used to build the shortest path tree. As shown in later experiments, it will significantly increase the detection rate when launching the basic WormCircle two or three times. An interesting research issue behind the basic WormCircle approach is how to coordinately select multiple roots to maximize the detection rate.

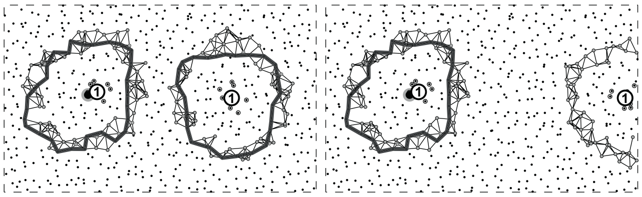
III. Localized WormCircle Algorithm

In the foregoing section, we describe the basic WormCircle and discuss its advantages and possible improvements on its shortcomings. The basic WormCircle identify the wormholeinfection symptom on a global shortest path tree, which make the detection effectiveness of basic WormCircle greatly depend on the location of the tree root. Certainly, as a global structure, the shortest path tree also needs the cooperation in the whole network. It is highly desirable to relax these limitations. In the section, we propose the improved method, called *localized WormCircle*. Localized WormCircle does not need to build a global shortest path tree, and make each node utilize only localized connectivity information to locate wormholes. As mentioned before, finding the proper wormhole symptom is the key to design a good countermeasure. Apparently, given local information, localized WormCircle only can detect localized deviation phenomenon in network topologies. Hence, the major challenges of this design lie in how to explore the local impacts caused by wormhole to characterize wormholes. Similarly, in the rest of this section, we first characterize the topological features of wormholes in the continuous domain, and then discuss its practical running in the discrete networks.

A. Locally Characterizing Wormholes

We classify wormholes into three categories in the the continuous domain, according to their topological impacts in the local. For α-class wormhole, both of its two endpoints locate inside the original surface. β-class wormhole has one endpoint inside original surface and the other on the boundary. γ-class wormhole has both its endpoints on the boundary. For example, the two wormholes shown in Figure 2 are both of α-class. Wormholes of different categories indeed imply the different difficulty to detect. Consider the scenario in Figure 6 (a). For wormhole labeled as 1, it is of α-class wormhole, and can be detected by WormCircle as long as the selected root is close to either endpoint. While for the β-class wormhole labeled as 2 to be detected, the root must be more close to the boundary endpoint in the upside. Thus, β-class wormhole is more difficult to detect than α-class for basic WormCircle. Clearly, a γ-class wormhole and a common bridge are indistinguishable from the local neighborhood. Hence, the γ-class wormholes cannot be detected with only using localized connectivity information accurately.

We mainly analyze the topological impact of the first two classes wormholes. Given the original surface S with wormholes, $d(x, y)$ denotes the geodesic distance between x and y in S. For an arbitrary point p in S and a small constant $\varepsilon > 0$, the ε closed neighborhood of p is denoted as $\varepsilon(p) = \{x \in S|\ d(x, p) \leq \varepsilon\}$. Further, we define the ε-*shell* of p as $\overline{\varepsilon}(p) = \{x \in S|\ d(x, p) = \varepsilon\}$. We then analyze how the different class of wormholes affects the topological structures of ε-shell of a point. Suppose there are no wormholes in S and let p be an inside point in S. Apparently, if ε is small enough, ε-shell of p will contain only one connected branch that is a cycle. Wormholes can change the structure of a ε-shell. Comparably, suppose that p locate at one endpoint of a α-class wormhole. It is clear to see that the small ε-shell of p will contain two cycles. In Proposition 1, we present more details about the local impact of wormholes in continuous domains. We can use Proposition 1 to detect wormholes in continuous topology surface, and each point can make decisions solely on its local information. That is the key idea of the localized WormCircle.



(a) (b)

Fig. 7. Detect the α-class wormhole in (a) and β-class wormhole in (b).

Proposition 1. *Let S be a plane region attached with one wormhole w, if there exist a point p in S and a positive constant ε, such that*

- *the ε-shell of p comprises two cycles, then w is a α-class wormhole and two endpoints of w locate in $\varepsilon(p)$.*
- *the ε-shell of p comprises only one cycle and contains tow or more connected branches, w is a β-class wormhole and two endpoints of w locate in $\varepsilon(p)$.*

B. Tracing Wormholes Locally

To implement the localized WormCircle in the discrete wireless networks, we also employ the node connectivity to verify the local continuity and utilize network hop count to approximate the geometric distance as in previous section. The main processes of localized WormCircle are as follows. Each node first obtains the connection relationship of its k hop neighbors. Each node then can locally detects whether there exists wormhole in its k hop neighbors according to the principles in Proposition 1. We use the instances in Figure 7 to illustrate the procedure, where the original network are randomly deployed on the rectangle region with 616 nodes and average degree 7.9. One α-class and one β-class wormhole are placed in the network in Figure 7 (a) and (b), respectively. We then discuss more details about the running of Localized WormCircle.

1) Procedures of Localized WormCircle:

Initially, each node obtains the connection relationship of its k hop neighbors. This can be achieved by exchanging neighboring list iteratively with constant times. k is constant and relies on the node distribution and topology density (generally it is set to 4 or 5 in our experiment). Once a node acquires the connection of its local neighbors, it selects the neighbors in the strip from l to k hops, $3 \leq l \leq k - 1$. Each node easily determines there are how many connected components in the strip. If there are two and more components, the node further validates whether some of these connected components form skeleton isoline circles in them. For each component, the node seeks a skeleton circle exactly as what we do for tracing isoline circles in the basic WormCircle. In the example of Figure 7 (a), the big dot node in Figure 7 (a) select its neighbors of 4 and 5 hops as the strip nodes, and discovers the strip forming two connected components. The big node further recognizes the two connected components

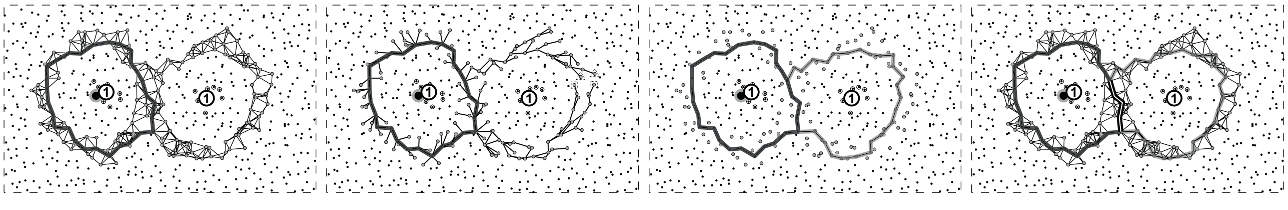


Fig. 8. Detect combined circles. (a) Primary circle, (b) Shortest path tree from primary circle, (c) (d) Find the combined circles.

as two discrete circles. The big node hereby reports that there is a α-class wormhole in local neighbors, according to Proposition 1. Similarly, the big node in Figure 7 (b) finds two connected branches and one circle, thus infers that one β-class wormhole exists among its neighbors. Note that if there are only one component, the node cannot just conclude that no wormhole in its local neighbors. It is different with the continuous case due to the network discreteness. We explain this case in the next.

2) Detecting Combined Circles:

We now explain why it needs further detection when there is only one connected component. We consider the problem shown in the Figure 8 (a), where there is one α-class wormhole. The thin lines denote the connection between the 4 and 5 hops neighbors of the big dot node. The expected two components are combined into one due to the relative closeness of the tow endpoints of the wormhole. Clearly, though there is only one connected component, the combined component is different with a discrete circle greatly. We hope our method is able to identify it as two *combined circles*.

For such the combined component, we can still find one circle as done in basic WormCircle. We select one shortest circle from all candidate circles, as the bold lines show in Figure 8 (a). We call the circle *primary circle*. To distinguish combined circles, we need to conduct the following extra steps. We perform a restricted flood from the primary circle in the combined component, and build a shortest path tree with multiple sources, the nodes in the primary circle, denoted by thin lines in Figure 8 (b). In the multiple-source shortest path tree, we try to find cut pairs using the same the criterion of cut pair as tracing wormhole circle in basic WormCircle. The triangles and squares in Figure 8 (b) denote the found nodes in the cut pairs. We can connect two cut pair nodes and trace back to their roots the primary circle in the multiple-source shortest path tree. Thus, we obtain a portion curve of the second circle, denoted as the light grey lines in Figure 8 (c), which adhering to the primary circle. Accordingly, the second circle is found successfully, and it shares a common segment with the primary circle, denoted by the double lines in Figure 8 (d).

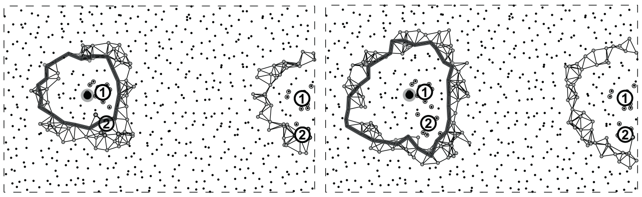
C. Discussion on Localized WormCircle

In the section, we present some discussion on localized WormCircle. The first problem involves how to locate the wormhole when multiple nodes all report the same wormhole. In this work, we do not focus on accurately differentiating and locating the wormhole link. When several nodes are close to the same endpoints of a wormhole, probably all detect the occurrence of the wormhole. The alerts for the wormhole can be aggregated and report that the small region is under wormhole attacks. This generally provides sufficient information for further response to a wormhole attack [11]. Moreover, there also exists the case that one wormhole can be reported as α-class by one node while is reported as β-class by other nodes. In such case, the wormhole preferably is reported to be of α-class. Note that it is not the objective of localized WormCircle to distinguish accurately the class of wormhole.

We mainly consider the solutions for one single wormhole in the before, since the algorithm is purely localized. When multiple wormholes emerge concurrently in the network, symptom of multiple wormholes may interfere with one another due to the network discreteness. We now analyze how the combination of multiple wormholes influences localized WormCircle. Apparently, if the endpoints of multiple wormholes distance each other far away, each one can be detected as a single wormhole respectively. However, when the endpoints are close to each other, the problem may become complex. Consider one example in Figure 9 (a), where exist two β-class wormholes. The endpoints of these two wormholes are close to each other. If there is no wormhole 2, the big node can find two connected components in its 2 to 3 hops neighbors, and detects wormhole 1 as β-class wormhole. However, the big node cannot detect wormhole 1 any more after wormhole 2 is added and connects the original tow components into one. Symmetrically, some nodes that originally are able to detect the wormhole 2 also fail due to the existence of wormhole 1. Intuitively, if the endpoints of wormholes are very close to each other, the impact of two wormholes should be regarded as one. Indeed, the tow wormholes can be detected as one β-class wormhole by some nodes, shown as the big node in Figure 9 (b). We say the two β-class wormholes are *coupled* with each other. Similarly, two α-class wormholes can also be coupled. Fortunately, it is not difficult to see that the coupled α-class wormholes can be tackled as detecting the combined circles. We omit more discussion due to space limitation.

IV. Performance Evaluation

In the section, we conduct extensive simulations under various situations to evaluate the effectiveness of our approach. By varying the node placement, node density, as well as the number and type of wormholes inside the network, we evaluate the success rate of detecting wormholes by the basic WormCircle (denoted as BW) and localized WormCircle (denoted as LW).



(a) (b)

Fig. 9. Impact of two close wormholes on localized WormCircle.

A. Simulation Setup and Evaluation Approach

In our simulations, the basic network setting is as follows. 6400 nodes are deployed in a 800*m* by 800*m* square area, and a single wormhole is attached in the network. We evaluate the algorithms with parameters in three orthogonal dimensions, *node distribution* model, *wormhole classification*, and *topology density*.

1) Node Distribution Model:

Nodes are deployed using two models in our simulations: *random placement* and *perturbed grid*. In the random placement model, the *x* and *y* coordinates of each node independently follows a uniformly random distribution on the region. Such a distribution models the network where nodes are randomly deployed throughout the field, e.g., dropping sensors from an airplane. Such a model contains irregularities in the network topology. The perturbed grid model deploys nodes on a grid and then perturbs each node with a random shift. We place nodes in a 80×80 grid, then shift each node with a random offset of at most one unit width. This model often is adopted to approximate manual deployments of nodes, corresponding more closely to planned organizations of a wireless network. It provides a uniform fill of sensors into the field.

2) Wormhole Classification:

As mentioned before, different classes of wormhole are of different difficulty to detect. We verify the effectiveness of our approaches about wormholes in varying classes. In each simulation, we randomly place a α-class or β-class wormhole with at least 4 hops span in the network. More concretely, the nodes in the network locating near on the borderline of square area are regarded as boundary nodes. One wormhole endpoint is considered to locate at the boundary of the network, if the allured nodes by the endpoint are distant to a boundary node within k hops, e.g. 3 hops. Otherwise, the wormhole endpoint is said to be inside the network. To construct a α-class wormhole, both its endpoints are positioned to the random places inside of the network. For β-class wormhole, one endpoint is randomly placed to be inside, and the other on the boundary.

3) Topology Density:

As remarked before, since the node density is an important factor in our algorithms. We use basic UDG model to build the network. Although our detection approach does not require any specific communication model for the network, UDG model is convenient to configure the network. We vary the communication radius of sensors from 13 meters to 26 meters, yielding average node degrees from 5 to 20.

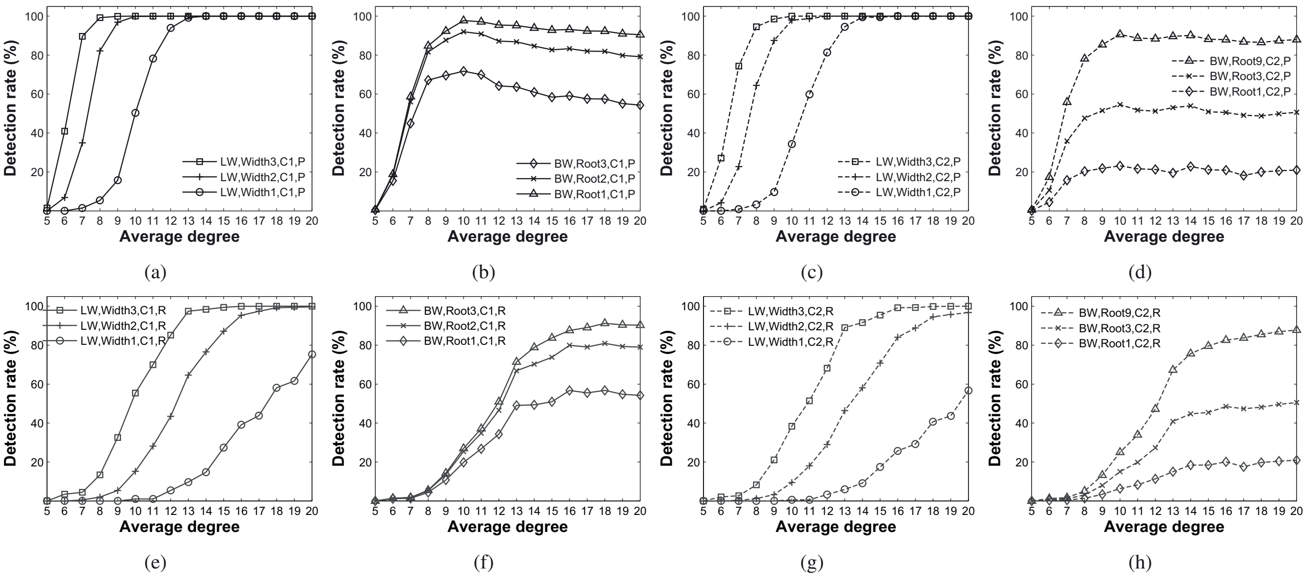
4) Evaluation Approach:

In the each simulation, after constructing the network according to above parameters, the basic WormCircle and localized WormCircle are launched to detect the wormhole respectively. As discussed before, using multiple roots in basic WormCircle would enhance the detection capability. Hence, we vary the number of roots in basic WormCircle to verify the improvement in each simulation. The isoline width is another important adjustable parameter for the both methods in the discrete network, since it used for tracing isoline in both basic and localized WormCircle. 1) For localized WormCircle, we change the isoline width from 1 to 3. Specifically, the neighboring nodes are selected in 3 hops, from 3 to 4 hops, 3 to 5 hops respectively. 2) For basic WormCircle, we just show the results when setting isoline width to be 3. For each set of simulation with fixed parameters, we compute and evaluate a detection result with repeating 500 times with random network generation and wormholes. Note that during our simulation we also test our approach on various network configures, such as adapting the model of quasi UDG, changing the fields of different shapes instead of square area, and obtain consistent results. We omit presenting the results due to the space limitation.

B. Analyzing The Result

Figure 10 shows all our performance results of basic WormCircle and localized WormCircle for two types of distribution models, two types of wormhole. In general, the following four groups of observations can be obtained from the results.

- *Effectiveness of WormCircle* Localized WormCircle provides very good results with 100% detection rate when the network with good connectivity. Basic WormCircle also achieves high ($> 80\%$) detection rate when with multiple roots. Localized WormCircle achieve more higher detection rate than basic WormCircle.
- *Impact of Node Placement.* Given other parameters, such as average node density, detection method, wormhole types, the detection performance gets worse as the randomness of node deployment increases, from comparing respectively the upper and lower four figures in Figure 10.
- *Impact of Different Types of Wormholes.* Given other parameters, the detection rate of α-class wormhole is bigger than β-class wormhole, from comparing respectively the left and right four figures in Figure 10.
- *Impact of Topology Density.* The detection rate generally increases as the node density increases. Nevertheless, we also notice the irregular phenomenon in Figure 10 (b) where the detection rate decreases as the node density increases when node degree is above 10. This mainly is because that when node density is relatively low, e.g. < 10, in perturbed grid model, the probability of successfully detecting wormhole circle increase greatly with the increase of node density. Nevertheless, when node density is relatively high, e.g. > 10, the probability of successfully detecting wormhole circle increase slowly with the increase of node density, since the successful



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 10. Wormhole detection rates for different configurations. P,R,$C1$,$C2$ in the legend indicate the distribution model of perturbed grid, random, α and β-class wormhole respectively. The upper four figures show the experiments where node distribution models are all of perturbed grid, while the lower four figures display the random case. The left four figures present the results for α-class wormhole, and the right four figures show for β-class wormholes. Width*N* in the legend denotes the isoline width of *N*. Root*N* means running basic WormCircle in *N* times.

probability has approached to 1. On the other hand, with the increase of communication radius, the hop distance from root to wormhole endpoints decreases. This leads to that the difference between the two distances from root to each wormhole endpoint decreases. Consequently, some wormhole circles may be indistinguishable or even disappear.

V. Conclusions

The wormhole attack is a severe threat to wireless ad hoc and sensor networks. Most existing countermeasures either require specialized hardware devices or have strong assumptions on the network, limiting their applicability in resource-constrained sensor networks. In this work, we explore the impact of wormhole attacks on the network topology, and develop two simple distributed detection methods, the basic and localized WormCircle. They rely solely on local connectivity information without any additional requirements on special hardware devices or making strong assumptions on network properties. WormCircle makes successful attempt to detect wormholes merely using local connectivity without any rigorous requirements and assumptions.

VI. Acknowledgments

This work is supported in part by the NSFC/RGC Joint Research Scheme N_HKUST 602/08, the National Basic Research Program of China (973 Program) under grant No. 2006CB303000, the National High Technology Research and Development Program of China (863 Program) under grants No. 2002AA1Z2101, No. 2007AA01Z177 and No. 2007AA01Z180, NSFC under grants No. 60621003 and No. 90718040. The authors are grateful for a variety of valuable comments from the anonymous reviewers.

References

[1] P. Papadimitratos and Z. J. Haas, "Secure routing for mobile ad hoc networks," in *Proc. of SCS CNDS*, 2005.

[2] K. Sanzgiri, B. Dahill, B. Levine, and E. Belding-Royer, "A secure routing protocol for ad hoc networks," in *Proc. of IEEE ICNP*, 2002.

[3] Y.-C. Hu, A. Perrig, and D. Johnson, "Packet leashes: A defense against wormhole attacks in wireless networks," in *Proc. of IEEE INFOCOM*, 2003.

[4] W. Wang, B. Bhargava, Y. Lu, and X. Wu, "Defending against wormhole attacks in mobile ad hoc networks," *Wiley Wireless Communications and Mobile Computing(WCMC)*, vol. 6, pp. 483–503, 2006.

[5] S. Capkun, L. Buttyan, and J.-P. Hubaux, "Sector: Secure tracking of node encounters in multi-hop wireless networks," in *Proc. of ACM SASN*, 2003.

[6] L. Hu and D. Evans, "Using directional antennas to prevent wormhole attacks," in *Proc. of NDSS*, 2004.

[7] I. Khalil, S. Bagchi, and N. B. Shroff, "Liteworp: A light-weight countermeasure for the wormhole attack in multihop wireless networks," in *Proc. of DSN*, 2005.

[8] ——, "Mobiworp: Mitigation of the wormhole attack in mobile multihop wireless networks," in *Proc. of IEEE SecureComm*, 2006.

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

[10] W. Wang and B. Bhargava, "Visualization of wormholes in sensor networks," in *Proc. of ACM WiSe*, 2004.

[11] R. Maheshwari, J. Gao, and S. R. Das, "Detecting wormhole attacks in wireless networks using connectivity information," in *Proc. of IEEE INFOCOM*, 2007.

[12] N. Song, L. Qian, and X. Li, "Wormhole attack detection in wireless ad hoc networks: a statistical analysis approach," in *Proc. of IEEE IPDPS*, 2005.

[13] L. Buttyan, L. Dora, and I. Vajda, "Statistical wormhole detection in sensor networks," in *Proc. of IEEE ESAS*, 2005.

[14] D. Huang, M. Mehta, A. van de Liefvoort, and D. Medhi, "Modeling pairwise key establishment for random key predistribution in large-scale sensor networks," *IEEE/ACM Trans. Netw.(ToN)*, vol. 15, no. 5, pp. 1204–1215, 2007.

[15] W. Gu, X. Bai, S. Chellappan, D. Xuan, and W. Jia, "Network decoupling: A methodology for secure communications in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.(TPDS)*, vol. 18, no. 12, pp. 1784–1796, 2007.

[16] Y. Wang, J. Gao, and J. S. Mitchell, "Boundary recognition in sensor networks by topological methods," in *Proc. of ACM MobiCom*, 2006.