Topological Detection on Wormholes in Wireless Ad Hoc and Sensor Networks

Dezun Dong, Mo Li, Yunhao Liu, Xiang-Yang Li, and Xiangke Liao

***Abstract*—Wormhole attack is a severe threat to wireless ad hoc and sensor networks. Most existing countermeasures either require specialized hardware devices or make strong assumptions on the network in order to capture the specific (partial) symptom induced by wormholes. Those requirements and assumptions limit the applicability of previous approaches. In this work, we present our attempt to understand the impact and inevitable symptom of wormholes and develop distributed detection methods by making as few restrictions and assumptions as possible. We fundamentally analyze the wormhole problem using a topology methodology, and propose an effective distributed approach, which relies solely on network connectivity information, without any requirements on special hardware devices or any rigorous assumptions on network properties. We formally prove the correctness of this design in continuous geometric domains and extend it into discrete domains. We evaluate its performance 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, target detection and tracking etc. Many of those applications involve a large number of sensing devices distributed in a vast geographical field to collaborate. Security is crucial for those mission-critical applications, which often work in unattended and even hostile environment. One of the most severe security threats [1] in ad hoc and sensor networks is wormhole attack, which has been independently introduced in previous works [2–4] and has spurred extensive research studies [5–17].

In wormhole attacks, the attackers tunnel the packets between distant locations in the network through an in-band or out-of-band channel. The wormhole tunnel gives two distant nodes the illusion that they are close to each other. The wormhole can attract and bypass a large amount of network traffic, and thus the attacker can collect and manipulate network traffic. The attacker is able to exploit such a position to launch a variety of attacks, such as dropping or corrupting the relayed packets, that significantly imperils a lot of network protocols including routing [4, 10], localization [18], and etc. This work focuses on typical wormhole attacks. The adversary is an outsider, who does not have valid network identity and does not become part of the network. The most severe feature of wormhole attack lies in the fact that the attacker can easily launch a wormhole attack without understanding the protocols used in the network or requiring compromising any legitimate node or cryptographic mechanisms. The attacker requires very little resources, i.e. a long-range directional wireless link, to replay packets verbatim. The establishment of wormhole attacks is independent of the general security mechanisms (in terms of confidentiality, integrity and authenticity of data) employed in the network. The attacker can forward each bit of a communication stream over the wormhole directly without breaking into the content of packets. Thus the attacker does not need to compromise any node and obtain valid network identities to become part of the network. The attacker with weak capabilities can launch an effective wormhole timely. Using the wormhole links, the attacker is able to gather enough packets and exploit the wormhole attack as a stepping stone for other more sophisticated attacks, such as man-in-the-middle attacks, cipher breaking, protocol reverse engineering, and etc. Wormhole attacks have posed a severe threat to wireless ad hoc and sensor networks.

The wormhole attack problem has received considerable attentions recently. Many countermeasures have been proposed to detect wormholes in wireless ad hoc and sensor networks. Those solutions typically catch the attacks by detecting partial symptoms induced by wormhole. Generally, existing symptom-based methods either depend on specialized hardware devices or make relatively strong assumptions on the networks. For example, some approaches employ specialized hardware devices, such as GPS [4, 8], directional antennas [5], or special radio transceiver modules [13], which introduce significant amounts of extra hardware costs for the systems. Other types of approaches are based on strict assumptions, such as global tight clock synchronization [4], special guard nodes [7], attack-free environments [14], or unit disk communication models [12]. These rigorous requirements and assumptions largely restrict their applicability in networks composed of a large number of low-cost resource-constrained nodes.

To fully address wormhole attack in ad hoc and sensor network, we need to answer the following two questions: (1) what symptoms feature the most essential characteristics caused by wormhole attacks and (2) how to gracefully design the countermeasures without critical requirements or assumptions. Our design goal is to rely solely on network connectivity information to detect and locate the wormholes. We focus our study on a fundamental view on the multihop wireless network topologies, aiming at catching the topological impact introduced by the wormhole. More concretely, we explore the fact that a legitimate multihop wireless network deployed on the surface of a geometric terrain (possibly with irregular boundaries, inner obstacles, or even on a non-2D plain) can be classified as a 2-manifold surface of genus 0, while the wormholes in the network inevitably introduce singularities or higher genus into the network topology. We classify wormholes into different categories based on their impacts on topology. We then design a topological approach, which captures fundamental topology deviations and thus, locates the wormholes by tracing the sources leading to such exceptions. Our approach solely explores the topology of the network connectivity. We do not require any special hardware devices, yet have no additional assumptions on the networks, such as awareness of node locations, network synchronization, unit disk communication model, or special guard nodes. The detection algorithm is carried out in a distributed manner across the network to avoid dependence on a small portion of the network, which could become the target of the adversaries. Although node density impacts on the detection performance of the method, our method works well in networks with fair node densities(e.g. node degree is greater than 7 in perturbed distributions and 16 in random distributions, respectively), which is verified by our simulations.

The rest of this paper is organized as follows. We first discuss those existing studies in Section II, and then formally define the wormhole problem and its detection methods in Section III. Section IV characterizes the wormholes in topologies and describes theoretical principles of a fundamental detection method. Section V presents our topological detection approach in discrete networks. We analyze the performance-cost tradeoffs and design light-weight approaches in Section VI. We evaluate this work through comprehensive simulations and analysis in Section VII. Finally we conclude this work in Section VIII.

II. Related Work

Existing countermeasures largely rely on observing the derivative symptoms induced by wormholes residing in the network. All of these approaches have their respective advantages and drawbacks. Applicability of approaches is largely dependent on specific system configurations and applications.

Some approaches observe the symptom of Euclidean distance mismatch in the network. Hu et al. [4] introduce geographic packet leash. By appending the location information of the sending nodes in each packet, they verify whether the hop-by-hop transmission is physically possible and accordingly detect the wormholes. Wang et al. [8] instead verify the end-to-end distance bounds between the source and the destination nodes. Zhang et al. [19] propose location-based neighborhood authentication scheme to locate the wormholes. Such approaches require the pre-knowledge of node locations to capture the distance mismatch.

Some approaches observe the symptom of time mismatch in packet forwarding. Hu et al. [4] introduce temporal packet leash, which assumes tight global clock synchronization and detects wormholes from exceptions in packet transmission latency. Capkun et al. [13] propose SECTOR which measures the round-trip travel time (RTT) of packet delivery and detects extraordinary wormhole channels. SECTOR eliminates the necessity of clock synchronization, but assumes special hardware equipped by each node that enables fast sending of one-bit challenge messages without CPU involvement. TrueLink proposed by Eriksson et al. [10] is another RTT based approach. It relies on the exchange of vast verifiable nonces between neighboring nodes. They modify the standard IEEE 802.11 protocols for the implementation. It remains unclear how effective such an approach is for the resource constraint ad hoc or sensor network hardware.

Some approaches observe the symptom of neighborhood mismatch that leads to physical infeasibility. Hu et al. [5] adopt directional antennas and find infeasible communicating links by utilizing the directionality of antenna communication. Khalil et al. [14] propose LiteWorp, which assumes the existence of an attack-free environment before the wormhole attacks are launched. During the deployment phase, each node collects its 2-hop neighbors and LiteWorp then selects guard nodes to detect wormhole channel by overhearing the infeasible transmissions among non-neighboring nodes. They further propose MobiWorp [15] to complement LiteWorp with the assistance of some location-aware mobile node.

Some approaches observe the symptom of graph mismatch under special assumptions of network graph models. Poovendran et al. [7][11] present a graph based framework to tackle wormholes. Their approach assumes the existence of guard nodes with extraordinary communication range. The direct communication links between guard nodes and regular nodes implicitly form a geometric graph and the wormholes will break the constraints. Wang et al. [6] graphically visualize the presence of wormholes. They reconstruct the layout of the networks by multi-dimensional scaling (MDS). Through the distance measurements between neighboring nodes, a central controller calculates the network layout and captures the wrap introduced by wormholes. Recently, authors in [12] propose a completely localized approach to detect wormholes with only network connectivity. By exploiting the forbidden packing number in the Unit Disk Graph (UDG) embedding of network graphs, the approach is able to detect wormholes with high accuracy. As a clear and elegant approach, however, it has its own limitations due to the assumption of UDG graph model and its basis on the symptom of packing number. It may fail when a wormhole does not cause an increase of packing number. It is thus inaccurate under non-UDG graphs.

Some approaches observe the symptom of traffic flow mismatch based on statistic analysis on the network traffic. Song et al. [16] observe the fact that the wormhole links are selected for routing with abnormally high frequency and by comparing with normal statistics they can identify the wormhole links. Another statistical approach proposed by Buttyan et al. [17] captures the abnormal increase of the neighbor number and the decrease of the shortest path lengths due to wormholes. The base station then centrally detects wormholes using hypothesis testing based on pre-statistics of normal networks.

To sum up, existing approaches heavily rely on specialized hardware or rigorous assumptions to capture the wormhole symptoms. Indeed, there are still no perfect symptoms found to establish an all-round method in the resource-limited ad hoc and sensor networks. Our design, based on topological observation, is orthogonal to existing approaches and takes a step towards relaxing these assumptions and expanding the applicability of methods.

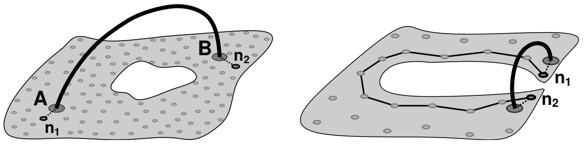
There is another completely different class of attacks than the classic wormhole attack, called *Byzantine wormhole* [20], which is a Byzantine variant of traditional wormhole attack. In a Byzantine wormhole attack, the attacker no longer come from outside the network, but from inside the network. The adversary has compromised one or more nodes, thus overwhelming the authentication-based security mechanisms. Insider attacks are more difficult to address since a compromised device can exhibit arbitrary malicious behavior. Awerbuch et al. [20, 21] propose a secure routing protocol, ODSBR. The goal of ODSBR is to provide routing survivability under Byzantine attacks, including black hole, flood rushing, and Byzantine wormhole attack etc. Khalil et al. [14] present LiteWorp, which uses local monitoring to address packet dropping and relaying in Byzantine wormholes. Eriksson et al. [22] propose a secure routing protocol, called Sprout, to defense multiple colluding insider attackers. Sprout is resilient to the shortcuts triggered by wormholes. More related techniques [23, 24] have also been proposed to address the issue of compromised nodes, including tamper-proof hardware, software tamper resistance and proofing, intrusion detection etc. The essence of those works, however, is to mitigate the malicious behaviors of wormhole attackers in several aspects, not to explicitly catch the wormholes.

III. Problem Formulation

In this section, we present wormhole attack model and system assumptions. We formulate the generalized wormhole problem with network connectivity.

A. Assumptions and Attacker Model

We consider a collection of homogeneous nodes deployed over a surface of terrain. Each node performs the homogeneous transmission control, and is only capable of communicating with adjacent nodes in its proximity. We do not force a unit disk graph communication model. Two nodes may or may not communicate with each other even their distance is within the maximum communicational range. We assume that the coordinates of nodes are unavailable, such that nodes can determine neither distances nor orientations of other nodes. In wormhole attacks, the attackers tunnel the packets between distant locations in the network through a high-speed out-of-band channel. The wormhole tunnel gives two distant nodes the illusion that they are close to each other. Figure 1 (a) displays a classic example of a wormhole attack. 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 [4]. 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.



(a) (b)

Fig. 1: Two examples of wormhole attack.

We make the common assumptions on wormhole attacks, which are widely adopted in most previous wormhole countermeasures [4–12]. Wormhole attacks are defined based on the minimum capabilities required by the attacker to perform these attacks. In particular, wormhole attacks are launched with mere hardware requirements. The attacker does not need to compromise any node, or have any knowledge of the network protocol used. Wormhole endpoints deployed by the adversary do not have valid network identities and do not become part of the network. The adversary launches *outsider* wormhole attacks in the network. We assume that in the network exist mechanisms that authenticate legitimate nodes and establish secure links between authenticated nodes. The communications can be protected by light-weight symmetric-key or asymmetric cryptographic mechanisms for sensor networks in link and upper layers[25, 26]. Confidentiality, integrity and authenticity of communications can be preserved in the networks under wormhole attacks. The adversary cannot fabricate and deliberately tamper with a message while escaping the detection of message authentication mechanism. The encrypted messages among valid network nodes keep confidential from the wormhole attacker, and thus the adversary can only drop and corrupt the relayed packets blindly. Those corrupted packets can be handled by techniques of reliable routing [27], such as retransmission on unstable links. To summarize, although wormhole attacks impact neighboring discovery mechanisms in the physical or link layer greatly, transmitted data over encrypted network protocols remains transparent and unobservable to the wormhole attacker, as formulated in most previous works [4–12].

B. Connectivity-Based Wormhole Problem

Poovendran et al. gave a formal definition of the wormhole problem based on the UDG communication graph model in Euclidean space [11]. According to their definition, a communication link is a wormhole link if the distance between its two endpoints exceeds the regular communication range. This concise definition, however, also has its own limitations. First, the definition is given under the constraints of the UDG communication graph model, which has been proven far from practical in many analytical and experimental works. Second, the distance-based definition in Euclidean space naturally binds the wormhole features with external geometric environments, and thus neglects the inherent topological impacts introduced by wormholes.

For example, consider the network shown in Figure 1 (b). The Euclidean distance between node n_1 and n_2 can be very 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 the external bold-line link 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. Obviously, in this case a wormhole attack occurs but it is not covered by the definition in [11], because the distance between nodes n_1 and n_2 does not exceed the maximum communication range. Such a wormhole attack cannot be detected by approaches based on Euclidean distance mismatch, as the geometric distance does not correctly reflect the network communication path. We hereby present a more general and fundamental definition of the wormhole attack based only on network topologies and aim to present the inherent characteristics of wormholes.

Definition 1: (*Generalized Wormhole Attacks*) Let G be a communication graph of a network, and w be an attack on the network. Let G_w be the perceived communication graph after the attack w. Let $L(u,v)$ and $L_w(u,v)$ denote the lengths of the shortest paths between an arbitrary pair of nodes $u, v \in V(G) \cap V(G_w)$ on G and G_w respectively. If $L_w(u,v) < L(u,v)$, we say that G_w is under wormhole attacks (or w launches a wormhole attack). $\lambda_{uv} = L(u,v) - L_w(u,v)$ quantifies the shortened path length of w between u and v. The intensity of the wormhole attack w is defined as $\lambda = max\{\lambda_{uv} | u, v \in V(G) \cap V(G_w)\}$.

Definition 1 formalizes the wormhole attack based only on the network topologies. The wormholes defined by Poovendran et al. are indeed all included by our definition. The attack intensity λ describes the intensity of the topological distortion brought by the wormhole attack. Intuitively, a larger λ corresponds to a more intensive distortion on network topologies. We then present our definition on generalized wormhole detection method.

Definition 2: (*Generalized Wormhole Detection Methods*) Let $\mathcal{G}_L \subseteq \mathcal{G}$ denote the set of legitimate network communication graphs, where $\mathcal{G}$ is the set of arbitrary communication graphs. Let $\mathcal{K}$ denote the pre-knowledge on legitimate network communication graphs. Let $\mathcal{P}$ denote the set of network properties, including graph or topological invariants. $\mathcal{M}_{\mathcal{K}} : \mathcal{G} \rightarrow \mathcal{P}$ is a mapping from the set of communication graphs to the set of network properties. If for any $G \in \mathcal{G}_L$, $\mathcal{M}_{\mathcal{K}}(G) \subseteq \mathcal{M}_{\mathcal{K}}(\mathcal{G}_L)$, $\mathcal{M}_{\mathcal{K}}$ provides a detection method, which does not cause false positive results. If for any graph $G \notin \mathcal{G}_L$, $\mathcal{M}_{\mathcal{K}}(G) \nsubseteq \mathcal{M}_{\mathcal{K}}(\mathcal{G}_L)$, $\mathcal{M}_{\mathcal{K}}$ is a detection method without false negative. $\mathcal{M}_K$ is a perfect method if it produces neither false negative nor false positive results.

Essentially, Definition 2 covers all possible methods that rely on network topologies for detecting wormholes. Different specific methods differ on assuming what pre-knowledge on the legitimate network and exploring what properties of the network topologies. For example, we explain this by an instance of wormhole detection methods which has been recently introduced by Maheshwari et al. [12]. Their method assumes a pre-knowledge $\mathcal{K}$ that the legitimate network communication graph is UDG, and mainly relies on the property $P \in \mathcal{P}$ that the lune packing number in an UDG embedding of the legitimate network communication graph is 2.

An ideal wormhole detection method should require as little pre-knowledge assumptions about the network as possible. The only pre-knowledge that we will assume is the fact that the network is deployed on a continuous geometric surface (2-manifold), where each node locally communicate with neighboring ones. We do not assume the availability of locations and Euclidean distance or time measures, yet we do not rely on any specific graph models like UDG or quasi-UDG for the network communications. In this paper, we deepen our study into a macroscopic view of the network topologies and characterize the essential impact of wormholes through. Deeply understanding of the topological impacts of wormholes, we accordingly propose the fundamental wormhole detection approach and analyze the performance-cost tradeoffs in topological wormhole detections.

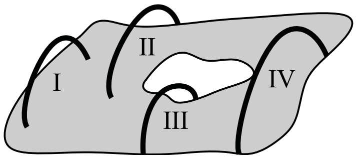


Fig. 2: Four different types of wormholes on the surface.

IV. Characterizing Wormholes

In this section, we model and characterize wormhole attacks on network topologies, and then propose the detection approach accordingly. Aiming at a distributed algorithm based on minimum assumptions on the pre-knowledge of a network, we intend to detect wormholes by solely depending on local cooperation and estimations. Nevertheless, the topological impact of wormholes is global, so how to characterize the global properties of wormholes from local information becomes a major challenge. We address the above problem through algebraic topology, by using homology and homotopy in general topological space. We introduce concepts, develop principles and present related theorems in continuous domain. We first introduce topological preliminaries. We then characterize the topological features of wormholes and classify the wormholes. Finally, we present the principles for the wormhole detection and prove theoretical guarantees. We extend our discussion to practical discrete networks in the next section.

A. Preliminaries

We use concepts and terminologies in combinatorial and computational topology. We first give a brief overview on the concepts and theories involved in our later discussions. Not all definitions are necessarily standard. For detailed explanations, see the books by Hatcher [28].

Given a topological space T, a *path* is a continuous function $p : [0,1] \rightarrow T$; a path whose endpoints coincide is called a *loop*. A *homotopy* between two paths p and q with the same endpoints is a continuous function $h : [0,1][0,1] \rightarrow T$, such that $h(0,t) = p(t)$ and $h(1,t) = q(t)$ for all t, and $h(s,0) = p(0) = q(0)$ and $h(s,1) = p(1) = q(1)$ for all s. Two paths are *homotopic* if there is a homotopy from one to the other. A loop is *contractible* if it is homotopic to a point.

In our work, we consider network deployment region as connected, compact and orientable (two-sided) 2-manifold *surfaces* that are topological Hausdorff spaces, where each point has a neighborhood homeomorphic either to the plane or to the closed half plane. This definition contains almost all ordinary surfaces observable in our daily life. In the rest of the paper, all *surfaces* mean such surfaces unless we explicitly

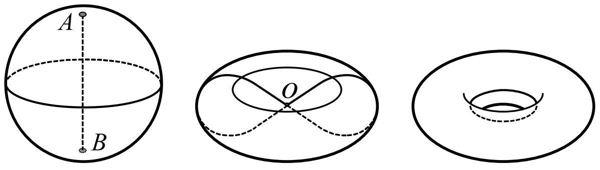


Fig. 3: (a) Link AB glued on a spherical surface X; (b) Link AB is contracted to a single point O; (c) Torus Y, which may collapse into $X \backslash AB$ by contracting a longitudinal cycle into one point.

state otherwise. When topological space T is a given surface S, a *curve* is a path and a *closed curve* is a loop. A *simple closed curve* is an injective closed curve that does not intersect itself. Two curves with the same endpoints on S are *homotopic* to each other if and only if one can be smoothly deformed to the other without leaving the surface. A closed curve is *contractible* if it is homotopic to a point, otherwise it is *non-contractible*. A closed curve is *non-separating* if the surface keeps connected after its removal. A closed curve is *separating* if it splits the surface into two or more components. The genus of a surface represents the maximum number of simple closed curves that can be removed without disconnecting the manifold. For example, a sphere and a disc have genus 0, while a torus has genus 1. Homotopy is actually an equivalence relation on the set of closed curves on S with any fixed basepoint. It classifies the set of cycles on a given surface into a set of homotopy classes, where cycles in each class are transformable to one another while cycles in different classes are not.

B. Characterizing Wormholes

Normally, a wireless multihop network is deployed on the surface of a geometric environment, such as a plane or a rough terrain. In this section, we develop principles in continuous domain, assuming continuous deployment of nodes over the geometric surface with one-to-one mapping to the points on the surface. In the continuous setting, a legitimate network is a 2-manifold surface without singular points and of genus 0, which is homotopic to the plane area with a certain number of boundaries (holes). We refer to the surface of the legitimate network as *original surface*. A wormhole link is a continuous line segment with extremely short length that connects two points on the surface.

A new topology space is formed after the wormhole is glued on the original surface. We subsequently analyze how the different topology spaces are generated after gluing different types of wormholes. We classify wormholes into four categories, according to their topological impacts. Figure 2 shows the four types of wormholes. For Class I wormhole, both of its two endpoints locate inside the surface. Class II wormhole has one endpoint inside the surface and the other on the boundary of the surface. Class III wormhole has its endpoints on two different boundaries. Class IV wormhole has both of its endpoints on the same boundary. The four types of wormholes have different topological impacts on the original surface, and the complex wormhole attack can be considered as a finite combination of them. We first consider the impact of a single wormhole. We then analyze the impact of the combination of multiple wormholes.

1) Single Wormhole Impact:

In this section, we analyze the impact of a single wormhole in different types, from Class I to IV. The main results are presented in Theorem 1.

Theorem 1: After inserting one wormhole into the original surface, Class I or II wormhole adds one degenerated genus, Class III wormhole adds one genus and reduces a boundary, and the Class IV wormhole adds a boundary.

Class I and II wormholes. Figure 3 shows an example of how a spherical surface X is affected by a wormhole link AB, which represents a Class I or II wormhole. Figure 3 (a) shows the new topology *quotient space* $X \backslash AB$ [28], with link AB glued on the spherical surface X. Figure 3 (b) shows a homotopy equivalent topology with Figure 3 (a), which contracts the line AB into a single point O. The new topology space can be considered as collapsed from a torus Y, as shown in Figure 3 (c). By contracting a longitudinal cycle around the torus, Y collapses into $X \backslash AB$. Clearly, such a collapse is not a homotopy equivalence from Y to $X \backslash AB$. In this sense, we say that $X \backslash AB$ contains degenerated genus 1. Strictly speaking, the new topology space after the injection of Class I or II wormhole is no longer a surface, as the neighborhood of the wormhole endpoint is not homeomorphic with a plane or closed half plane. Informally, we call it as a surface with singularities.

Class III wormholes. When the surface is of multiple boundaries (the network containing physical holes), Class III wormhole might appear as shown in Figure 4 (a). The topology space of Figure 4 (a) is homotopy equivalent to that in Figure 4 (b), which contracts the wormhole link into a point. We focus on the two non-contractible cycles α and β in Figure 4 (b). Cycle α goes through the wormhole, and cycle β wraps the inner boundary. Figure 4 (b) can be seen as the deformation retract of Figure 4 (c), where the cycles α and β in Figure 4 (c) correspond to α and β in Figure 4 (b) respectively. Indeed, Figure 4 (a-c) are homotopy equivalent to each other. Typically, a Class III wormhole concatenates two different boundaries and increases the genus by 1. An interesting phenomenon happens under Class III wormhole. The twisted cycle α and cycle β are actually symmetrical to each other in the sense of topology. Imaging that if we overturn the surface in Figure 4 (c), the meridional circle α becomes a longitudinal circle, while the longitudinal circle β becomes a meridional circle. Without the knowledge that β is homotopic to a physical boundary beforehand, we are not able to differentiate α and β in Figure 4 (b) through only topologies.

Class IV wormholes. A Class IV wormhole connects two points on the same boundary. Thus Class IV wormhole adds a bridge to the original surface and separates the boundary into two.

In summary of above discussions, we obtain the Theorem 1.

2) Combination of Multiple Wormholes:

When two or more wormholes exist on the surface, Class I or II wormholes still introduce independent impacts, each leading to the increase of degenerated genus by 1. Multiple

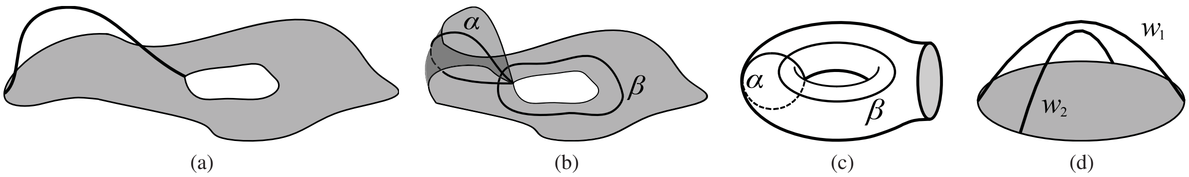


Fig. 4: The impact of wormholes. (a) A single Class III wormhole; (b) The homotopic surface when contracting the wormhole link in (a); (c) The homotopic surface to (a) and (b); (d) Two Class IV wormholes crossing each other.

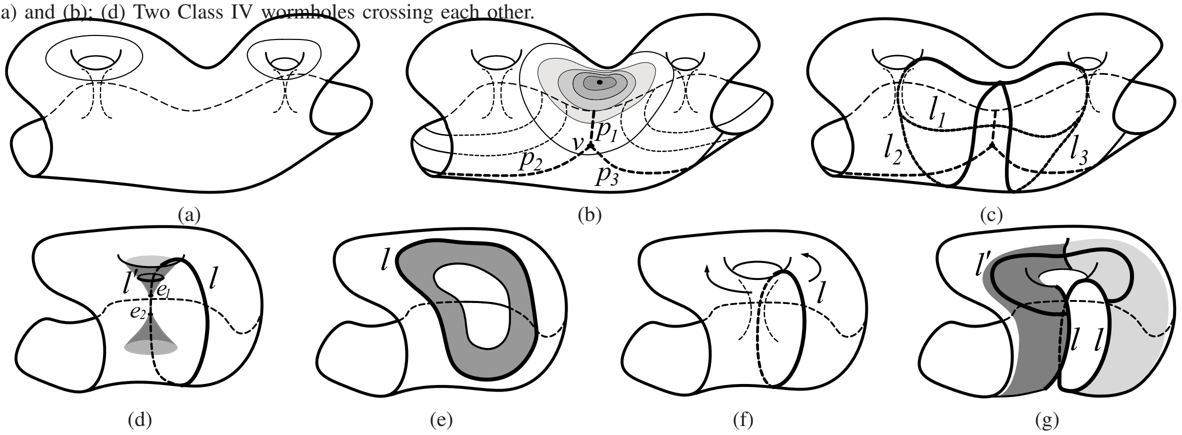


Fig. 5: Tracing wormholes by topologies.

Class III and Class IV wormholes, however, might introduce interchangeable effects. As the example shown in Figure 4 (d), two Class IV wormholes w_1 and w_2 are injected on the surface crossing each other. A single wormhole w_1 or w_2 adds a boundary to the surface, but the combination of them adds genus by 1. As a matter of fact, Figure 4 (d) is homotopy equivalent to Figure 4 (a-c). The example above can be explained as follows. After the first Class IV wormhole w_1 or w_2 is glued on the surface, the boundary of the original surface is split into two. When we add the second Class IV wormhole, its two endpoints are then on two different boundaries, so the wormhole is slid to a Class III wormhole to the new surface. The consequence is a combination of a Class IV wormhole and a Class III wormhole, leading to the increase of genus.

When multiple wormholes are injected to the original surface, we can consider them as being sequentially glued to the surface. The type of each wormhole is determined according to the instant surface when it is glued. Class I and II wormholes will not be affected by previous injected wormholes, while Class III and IV wormholes might interchange their types according to the boundary separation or concatenation. The sequence in gluing the wormholes does not affect the final topological impact. We look into the final impact of multiple wormholes and characterize the topology surface with genus g, degenerated genus d and b boundaries as $\tau(g,d,b)$, where g, d and b are non-negative integers.

Theorem 2: Given the original surface $\tau_0 = \tau(g_0, d_0, b_0)$ and the final surface $\tau(g,d,b)$ after N wormholes are injected, there is $N = 2(g-g_0) + (d-d_0) + b - b_0$. Among the N wormholes, there are $d - d_0$ Class I or II wormholes and $2(g-g_0)+b-b_0$ Class III or IV wormholes.

Proof: The proof is by induction on the number N of wormholes. Let $\tau_i = \tau(g_i, d_i, b_i)$ denote the intermediate surface after adding i wormholes on τ_0. Without losing generality, we add the N wormholes sequentially. We denote the sequence as $[w_1, w_2, \cdots, w_N]$. When $i = 1$, there is a single wormhole added to the network. It is clear from Theorem 1 that this theorem is true. Assume the theorem is true for $i = k$, i.e., in current surface $\tau_k = \tau(g_k, d_k, b_k)$, there are $d_k - d_0$ Class I or II wormholes and $2(g_k - g_0) + b_k - b_0$ Class III or IV wormholes to τ_0. When a new wormhole w_{i+1} is added, the surface becomes $\tau_{k+1} = \tau(g_{k+1}, d_{k+1}, b_{k+1})$. There are three cases: 1) if w_{i+1} is a Class I or II wormhole to τ_0, it is still a Class I or II to τ_k. Thus, we have $g_{k+1} = g_k$, $d_{k+1} = d_k + 1$, $b_{k+1} = b_k$. So the number of Class I and II wormholes to τ_0 is $d_k - d_0 + 1 = d_{k+1} - d_0$, by the induction hypothesis, the number of Class III and IV wormholes to τ_0 is still $2(g_k - g_0) + b_k - b_0 = 2(g_{k+1} - g_0) + b_{k+1} - b_0$. 2) if w_{i+1} is a Class III wormhole to τ_0, it might be a Class III or IV wormhole to τ_k. In the previous case, we have $g_{k+1} = g_k + 1$, $d_{k+1} = d_k$, $b_{k+1} = b_k - 1$. In the latter case, we have $g_{k+1} = g_k$, $d_{k+1} = d_k$, $b_{k+1} = b_k + 1$. In either case, the number of Class I and II wormholes to τ_0 is $d_k - d_0 = d_{k+1} - d_0$, and the number of Class III and IV wormholes to τ_0 is $2(g_k - g_0) + b_k - b_0 + 1 = 2(g_{k+1} - g_0) + b_{k+1} - b_0$. 3) if w_{i+1} is a Class IV wormhole to τ_0, it is similar with case 2. Thus, this theorem is true for the case of $i = k+1$. ■

According to our per-knowledge on the legitimate network graph, the original surface has genus 0 and degenerated genus 0, so the original surface can be characterized as $\tau(0, 0, b_0)$ where b_0 is the number of boundaries (which is equal to the number of inner holes + 1). According to Theorem 2, we can calculate the number of different types of wormholes if we can characterize the final topology space.

C. Tracing Wormholes

We hereby present the principle of tracing wormholes in continuous topology surface. For the convenience of presentation, we take a macroscopic view on the global network. In real implementation, the algorithm does not depend on centralization throughout the network. A node makes decisions solely based on its local information. We use an example of a surface with wormholes shown in Figure 5 to explain this design. The proposed algorithm aims to trace wormholes through detecting the genus and degenerated genus. The main idea of the algorithm is to find the non-separating cycles associated with wormholes. Two circular lines in Figure 5 (a) indicate two potential non-separating cycles in this example. The algorithm is described in Algorithm 1.

1) Finding Cut Locus and Candidate Loops:

Given the wormhole infected surface S, we first select an arbitrary point in S as the root and run a continuous *Dijkstra* shortest path algorithm [29], as shown in Figure 5 (a). Each point is thereafter aware of its shortest geodesic paths to the root. We call the set of points that have more than one shortest path to the root the *cut locus* [29], denoted by C_S. After discovering the *Dijkstra* shortest paths to the root, we find a cut locus forms there. If we cut the surface along the cut locus, the surface becomes a topological disk. The paths marked by bold dashed lines are part of the cut locus. The point in cut locus which has at least three shortest paths to the root is called a *branch vertex* of the cut locus, like point v in Figure 5 (b). The branch vertices separate the cut locus into *cut paths*, like path p_1, p_2 and p_3 in Figure 5 (b). Each cut path has two endpoints. The endpoint of a cut path can be a branch vertex or not. We call the endpoint *leaf vertex*, if it is not a branch vertex. The leaf vertex can be on the boundary or in the interior of the surface. We further distinguish them as *boundary leaf vertex* and *interior leaf vertex*. We can transform the cut locus C_S into its subgraph *reduced cut locus* through repeatedly removing all interior leaf vertices [29]. We denote the obtained reduced cut locus as $C(P, V)$, where P is the set of cut paths and V is the set of branch and boundary leaf vertices.

Let $p \in P$ be a cut path in the reduced cut locus and $a \in p$ be an arbitrary point on p. There are at least two non-homotopic shortest paths from a to the root. By concatenating the two non-homotopic paths, we obtain a loop l_a and it is clear that loop l_a is non-contractible. We say that a is the witness of l_a. For any two points $a, b \in p$, if l_a and l_b are the loops witnessed by a and b respectively, l_a and l_b are homotopy equivalent [29]. For each cut path $p \in P$, we arbitrarily select a loop witnessed by one point p and denote it as l_p. Thus we obtain a set of loops $L = \{l_p | \, p \in P\}$, which we call the *candidate loop set*. Figure 5 (c) displays the three candidate loops l_1, l_2 and l_3, corresponding to the three cut paths p_1, p_2 and p_3 in Figure 5 (b) respectively. Following Lemma 4.2 in [30], there are at most $4(g+d)+2b-2$ branch vertices, and $6(g+d)+3b-3$ cut paths. Hence, the number of candidate loops $|L| < 6(g+d)+3b-3$. For each candidate loop $l \in L$, we do the following steps to clarify the situations of wormholes.

2) Locating Class I or II Wormholes:

To begin with, for checking whether or not the loop passes through a degenerated genus (Class I or II wormholes), we consider a small closed ε-neighborhood $N(l)$ of l. $N(l) = \{\varepsilon(x) | \, x \in l\}$, where $\varepsilon(x)$ denotes the ε-neighborhood of point x on the surface. As shown in Figure 5 (d), the bold line denotes the candidate loop l, which passes through a Class I wormhole with its two endpoints labeled as e_1 and e_2. If there exists a sufficiently small simple closed curve l' in $N(l)$ that crosses l odd times (two curves are not crossed if they touch [28]), l can be marked as a loop through Class I or II wormhole. We call l an *independent non-separating loop*. We can further contract the cycle l' in the figure as much as possible while keeping it crossing l odd times. The cycle l' eventually contracts to one endpoint of the wormhole, i.e., node e_1 in Figure 5 (d). By this means, we can detect the endpoints of all Class I and II wormholes.

3) Detecting Class III or IV Wormholes:

The case of Class III and IV wormhole is different. As both endpoints of such wormholes are on the boundaries of a surface, we cannot find such a small cycle enclosing each endpoint of a wormhole. Instead, we directly detect the genus by checking whether the candidate loop l is a separating or non-separating loop. There is an essential difference between the two types of loops. The separating loop is two-sided but the non-separating loop is one-sided. Figure 5 (e) displays a separating loop that is formed due to the plain holes on the surface. It is two-sided in the sense that if we flood from the loop with different colors, e.g., the two colors never meet. The loop shown in Figure 5 (f), however, is a non-separating loop formed by genus. If we flood light grey and dark grey to its two sides, as shown in Figures 5 (f) and (g), the two colors ultimately meet with each other because the loop is one-sided. By detecting the non-separating loop l, we detect the genus introduced by Class III or Class IV wormholes. Let t be a point on the cut between the two color areas. Let $s \in l$ be an arbitrary point on l. There is a pair of non-homotopic paths from s to t, one across the area of one color and the other one across the other color area. The two paths form a loop, which we denote in Figure 5 (g) as l'. Apparently, l' crosses l at a single point s. As we will later see in Lemma 4, both l and l' are non-separating loops. We call l a *dependent non-separating loop* and l' the *partner loop* of l. Further, we call the two non-separating loops that cross each other *knit non-separating loop pair*. We can conclude that there must be at least one Class III or IV wormhole in the knit non-separating loop pair. Yet as we mention in Figure 4 (c), the two loops are topologically indistinguishable and we cannot conclude which loop passes through the wormhole.

To summarize, for each candidate loop $l \in L$, we classify it into one of the three types: separating loop, independent non-separating loop, or dependent non-separating loop. We detect and locate Class I and II wormholes from independent non-separating loops. We detect Class III and IV wormholes from dependent non-separating loops.

D. Correctness and Optimality

We prove that our method is able to detect all the detectable wormholes correctly. We first discuss the correctness and capability of this method, and then analyze the theoretical bound in topologically detecting wormholes.

Theorem 3: Let L be the set of candidate loops, all wormholes reside within L.

Proof: It is not difficult to prove that there exists a subset $L' \subseteq L$, which constitutes a homotopy basis of the original surface [29]. Let w be an arbitrary wormhole on the surface, and l_w is an arbitrary loop on the surface that passes through w. Since L' is a homotopy basis, there must exists a loop l_c

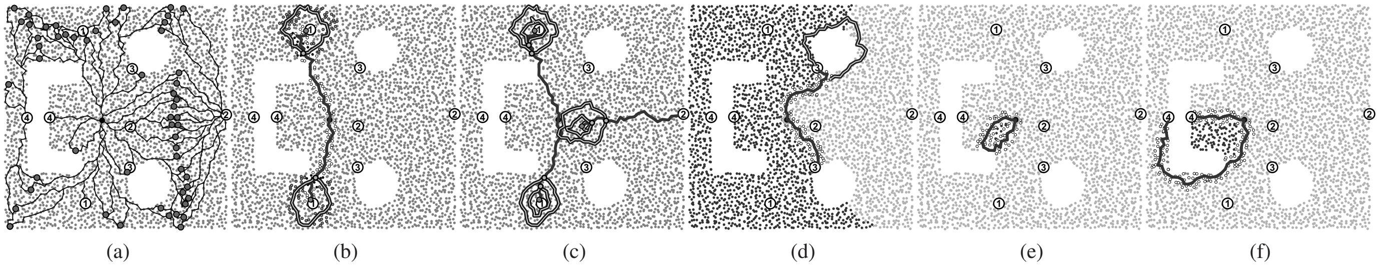


Fig. 6: Wormhole detection in discrete environments.

Algorithm 1 Tracing Wormholes in Continuous Domain

```
Input: The surface $S$, i.e. original surface attached with wormholes.
Output: The location of wormholes.
Select a root $r \in S$ to run a continuous Dijkstra shortest path algorithm in $S$; Obtain the cut locus $C_S$ of $S$.
Recursively remove interior leaf vertices of $C_S$, and transform $C_S$ into the reduced cut locus $C(P, V)$; Candidate loop set $L = \varnothing$.
for Each cut path $p \in P$ do
    Randomly select a point $a \in p$; Concatenate two non-homotopic paths from $a$ to root $r$, and obtain the loop $l_a$; $L := L \cup l_a$.
end for
for Each candidate loop $l \in L$ do
    if Successfully find a contractible cycle $l'$ in the $\varepsilon$-neighborhood $N(l)$ of $l$, which $l'$ crosses $l$ odd times then
        With keeping cycle $l'$ crossing $l$ in odd times, contract $l'$ into a point $o$; Report a Class I or II wormhole locates at $o$.
    else
        Flood two colors in the two sides of $l$.
        if Two colors meets and find a partner loop $l'$ of $l$ then
            Report that there is at least one Class III or IV wormhole in the knit non-separating loop pair $(l', l)$.
        end if
    end if
end for
```

homotopy equivalent to l_w while l_c can be represented as the concatenation of some proper loops in L'. It means w must be passed through by at least one loop in $L' \subseteq L$. ■

From Theorem 3, we have confined the locations of all possible wormholes within the candidate loops L, although we may not be able to locate exactly the endpoints of all wormholes on L. Now, we prove our method is effective and accurate on detecting Class I and II wormholes. We first present Lemma 4, which reveals the parity property of the non-separating loops.

Lemma 4: On surface S, a cycle c is non-separating if there is a cycle c' such that c' crosses c odd times.

Proof: Following Lemma 2.1 in [31], if c is separating, $S - c$ has two components S_1 and S_2, each with c as its boundary. If we trace the curve c', it must switch between S_1 and S_2 each time it crosses c, and never otherwise. Hence there must be an even number of switches, contradicting the fact that c and c' cross oddly. ■

Theorem 5: All Class I and II wormholes are detected and exactly located by our method.

Proof: Let w be an arbitrary Class I or II wormhole. According to Theorem 3, there exists a loop $l_w \in L$ which passes through w. Since w is a Class I or II wormhole, w increases one degenerated genus on the surface. For the degenerated genus, there exists a contractible simple closed curve at one end of the genus that crosses l_w one time, i.e., all Class I and II wormholes can be effectively detected without false negative. On the other hand, let l be an arbitrary loop in L. If there exists a contractible loop l' in the ε-neighborhood of l crossing l oddly, according to Lemma 4, l must be non-separating. l' is both non-separating and contractible, so l' is continuously deformed and contractible to an endpoint of at least one degenerated genus, never otherwise. When ε is sufficiently small, it guarantees that there is only one endpoint inside l'. Thus the detection method accurately locates the Class I and II wormholes. ■

Theorem 6: Let l and l' be a pair of knit non-separating loops. There is at least one Class III or IV wormhole on l and l'.

Proof: Suppose that neither l nor l' passes a wormhole, then l and l' are also loops on the original surface without wormholes. Since l and l' form a knit non-separating loop pair, l and l' cross in odd times, thus l and l' are both non-separating according to Lemma 4. On the other hand, since the original surface is homotopic to a plane area with holes, according to Jordan Curve Theorem [28], a loop in the original surface must separate the original surface into at least two components. Hence, both l and l' are separating, which leads to contradiction and finishes this proof. ■

Theorem 6 shows that our detection method is accurate on Class III and IV wormholes, i.e., each pair of knit non-separating loops captures at least one Class III or IV wormhole. We successively show by Theorem 7 and 8 that our method detects all topologically detectable wormholes on the original surface.

Theorem 7: The instant Class IV wormhole is homotopy equivalent to a plain bridge on previous surface, and thus is undetectable with topological method.

Proof: As we characterize in Section IV-B, an instant Class IV wormhole adds a bridge on the same boundary. In the sense of homotopy equivalence, it is indistinguishable with a plain bridge on previous surface. Thus Class IV wormhole is undetectable with topological method. ■

Theorem 8: Given the original surface $\tau_0 = \tau(0, 0, b_0)$, and the surface $\tau(g, d, b)$ after wormhole attacks. Our method locates all d Class I and II wormholes and detects at least g Class III or IV wormholes while the rest of wormholes are topologically undetectable.

Proof: First, according to Theorem 5, our method is able to locate all d Class I and II wormholes exactly. Second, according to Theorem 6, we can detect at least g Class III or IV wormholes by detecting g non-separating loop pairs for genus g. Third, we consider an arbitrary order of inserting the wormholes into the network. According to Theorem 1 and 2, an increase of genus happens when and only when instant Class III wormholes (might be Class IV to the original surface) are inserted. While the genus is increased by g, there are $g+b-b_0$

instant Class IV wormholes inserted. According to Theorem 7, their topological impacts on the network are indistinguishable from bridges and thus topologically undetectable. ■

To summarize this section, we introduce concepts, develop principles and present related theorems in continuous domain. We first introduce topological preliminaries. We then characterize the topological features of wormholes and classify the wormholes. Finally, we present the principles for the wormhole detection and prove theoretical guarantees. We extend our discussion to practical discrete networks in the next section.

V. Wormhole Detection In Discrete Environments

We have characterized the impact of wormholes and described the principles of wormhole detection under continuous settings in the previous section. In a real multi-hop network, however, nodes are deployed discretely on the field. In this section, we present our approach in discrete environments. First, we construct a shortest path tree from an arbitrarily selected root node, so that each node obtains shortest paths to the root. We accordingly select the candidate loops from the cut pairs on the shortest path tree. Second, we detect and locate Class I or II wormholes by testing whether a candidate loop is an independent non-separating loop. Specifically, we check whether there exists a contractible cycle that crosses the loop one time. Third, we check the existence of Class III or IV wormholes by seeking the knit non-separating loop pairs. All operations are carried out in a distributed manner in the discrete network. The principle of this design follows what we introduced in the continuous settings. When applied in discrete environment, however, there exist substantial technical challenges in transforming the principles into concrete protocols as follows. (1) It is non-trivial to test in discrete networks whether or not a cycled path is contractible, especially with only connectivity information among local neighborhoods. (2) Determining the crossing of two curves without any geometric information is challenging. To calculate the accurate crossing times of the two curves is even more difficult. (3) To seek the knit non-separating loop pairs, we need to check whether a candidate loop is one-sided or two-sided. Having solely the connectivity information, to determine the two sides of a path is also difficult.

We address above challenges in this design, which includes three components: *Candidate Loop selection*, *Finding Independent Non-Separating Loops*, and *Seeking Knit Non-Separating Loop Pairs*. We illustrate the operations using the example shown in Figure 6, where we have all four different types of wormholes residing in a network, denoted from 1 to 4.

A. Candidate Loop Selection

After the shortest path tree is established, each node knows its shortest paths to the root node. The neighboring nodes exchange the information of their shortest paths. There are some pairs of nodes connected with each other but with their least common ancestor far away. These nodes form *cut pairs* [32]. The cut pairs witness the candidate loops. The two shortest paths from the cut pair constitute a loop and we qualify a candidate loop by setting a threshold on the length of the loop. The threshold depends on the expectation of the span of wormhole attacks, i.e., if we aim to detect all wormholes across h hop span, we can set the threshold to h hops.

Figure 6 (a) plots the detected cut pairs (big nodes) and corresponding candidate loops (thin line paths). The shortest path tree is constructed by flooding from the big root node in the center. As shown in this example, there are variations on the candidate loops, including misreported ones. Due to the randomness and discreteness of the network deployment, it is indeed difficult to obtain the cut locus accurately under discrete settings. To tackle this problem, we perform all consecutive operations on all candidate loops, instead of selecting only one loop for each cut path as in continuous principles. Such operations might introduce extra network cost. In practice, we can filter most of redundant candidate loops simply by checking their neighboring relationship, which leads to significant savings on the overhead.

B. Finding Independent Non-Separating Loops

Let l denote a candidate loop. To test whether l passes a Class I or II wormhole, we verify whether or not l is an independent non-separating loop. As described in previous section, we need to find a small contractible circle that crosses l one time.

We articulate the concept of contractible circle in discrete settings. Given the communication graph G, and two positive integers k and δ. For a vertex $v \in V(G)$, let $\Gamma_k(v)$ denote the set of nodes within k hop distance to v. Let $\Gamma_{k,\delta}(v) = \Gamma_{k+\delta}(v) - \Gamma_k(v)$. Given a vertex set $U \subseteq V(G)$, let $G[U]$ denote the vertex induced subgraph of G from U. Thus, for an arbitrary node $v \in V(G)$ and $r, \delta \in \mathbb{N}$, if $G[\Gamma_{r,\delta}(v)]$ is a connected circular strip, we find a skeleton circle within $G[\Gamma_{r,\delta}(v)]$, as shown in Figure 7 (a). Tracing such a skeleton circle is non-trivial. We conduct a restricted flooding from an arbitrary node in the strip graph $G[\Gamma_{r,\delta}(v)]$ and build a shortest path tree, as shown in Figure 7 (b). The big circle is the selected root node and the lines show the shortest path tree. In this shortest path tree, we find an arbitrary cut pair among the leaf nodes depicted as triangle and square nodes in Figure 7 (b), and connect them to form a loop, similarly as what we do for constructing foregoing candidate loops. The dotted line connects the two cut pair nodes, the triangle and square nodes. We can thus trace back from the two cut pair nodes to obtain a candidate cycle, denoted by the grey lines in Figure 7 (c). We record it as $C(v, r, \delta)$. Apparently, when r and δ are sufficiently small, $C(v, r, \delta)$ is contractible. Moreover, we say that $\Gamma_k(v)$ is a k-hop contractible disk at v, if for any $r_0 \leq r \leq k$, there exists a skeleton circle within $G[\Gamma_{r,\delta}(v)]$. A contractible disk represents a set of network nodes embedded in a geometric region without voids and the skeleton circles on different levels of the contractible disk are all contractible circles. In our later example and simulations, we set $r_0 = 1$, $k = 3$ and $\delta = 2$.

By creating a contractible disk, we explore the existence of contractible circle $C(v, r, \delta)$ around each node v in the candidate loop l. If there exists such a circle $C(v, r, \delta)$, there must be intersection between $C(v, r, \delta)$ and l. In the discrete

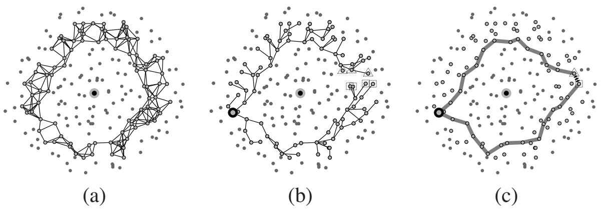


Fig. 7: Tracing a skeleton circle. (a) connected circular strip $G(\Gamma_{k,\delta}(v))$, (b) shortest path tree, (c) a candidate circle.

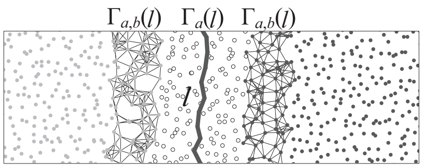


Fig. 8: Distinguishing the two sides of loop l.

settings, however, with only network connectivity information, it is yet challenging to determine how many times $C(v, r, \delta)$ crosses l. The two general curves might intersect with no common nodes or even at multiple ambiguous intersection nodes. Similar problems are also considered in [33]. Fortunately, we can restrictively transform our case into a relatively easier one, as we only need to judge if $C(v, r, \delta)$ crosses l once or not. We let $\Gamma_1(C)$ and $\Gamma_1(l)$ denote the sets of nodes within one hop distance to $C(v, r, \delta)$ and l respectively. Let $I = \Gamma_1(C) \cap \Gamma_1(l)$. We check if there is only one single connected component in I or not and accordingly conclude if $C(v, r, \delta)$ crosses l only in one time. We confirm that the candidate loop l is an independent non-separating loop if our test shows that $C(v, r, \delta)$ crosses l one time. Thus there must be one endpoint of the wormhole included in $C(v, r, \delta)$. Figure 6 (b) illustrates that our approach works on a candidate loop across a Class I wormhole. The vertical single line represents the candidate loop that passes through the wormhole. The double-line paths are the detected contractible circles that cross the candidate loop one time. The circles nodes that filled with white and grey are the one-hop neighborhoods of the single-line and double-line pathes, respectively. The dark dot nodes show the intersection set of the two kind of filled circle nodes. By shrinking the contractible circles, we can eventually locate the wormhole endpoints. As shown in Figure 6 (c), this approach successfully finds the contractible circles and locates the two endpoints of the Class I wormhole and one endpoint of the Class II wormhole. By tracing the traffic flow from one end, we can successively locate the other end of the Class II wormhole.

C. Seeking Knit Non-Separating Loop Pairs

To detect Class III or IV wormholes, we continue to test whether a candidate loop l passes through a Class III or IV wormhole. According to the principles in continuous case, we seek the knit non-separating loop pair containing l.

The principle is simple, i.e., we conclude whether loop l is separating or non-separating by checking whether l is one-sided or two-sided. This can be easily achieved in continuous settings by flooding two colors from l to its two sides and checking whether the two colors ultimately meet with each other. In discrete settings, however, it becomes difficult, as with only network connectivity information, we cannot distinguish the two sides of l. We cannot locally determine a node is on which side of l by solely connectivity.

We propose corresponding countermeasures to address the issue above. We first flood from loop l and construct a shortest path tree rooted at l. Each node is thus aware of its shortest distance to l. $\Gamma_a(l)$ denotes the set of nodes within a hop to l. Indeed, as Figure 8 shows, we let nodes in $\Gamma_a(l)$ keep silent, separating the shortest path tree into two parts corresponding to the two sides of l. We let each node within $\Gamma_{a,b}(l)$ delivers its specific color down to successive nodes. The color is represented by its node ID or a randomly generated number. The color value is first flooded within $\Gamma_{a,b}(l)$. During flooding, the smallest color value suppresses other color values. Then along the shortest path tree, the dominant color value is delivered and inherited by every node. In our implementations, we set $a = 2$ and $b = 4$. After the colors spread over the network, different colors classify the nodes in the network into at least two types, as Figure 6 (d) shows. We then verify whether the nodes with different colors neighbor to each other by exchanging the color information among neighboring nodes. If there does exist such a pair, loop l is one-sided. There are two paths from the pair of nodes to loop l through the two components of different colors, and accordingly the two paths can constitute a loop l'. l and l' compose a knit non-separating loop pair, as the pair of single-line and double-line loops found in Figure 6 (d). We then conclude that there is at least a Class III or Class IV wormhole on l or l'.

Figure 6 (e) displays the testing result against a separating loop formed in Figure 6 (a) due to the noise in finding accurate candidate loops. The loop separates the network into two parts, which confirms to be a two-sided. The loop, however, differs from the loops in Figure 6 (d), and it can be verified to be contractible loop by the local communication. Thus there will be no wormhole reported in Figure 6 (e).

Figure 6 (f) displays a candidate loop formed by a Class IV wormhole. As such a Class IV wormhole is topologically indistinguishable from a bridge across the void hole, the loop is also tested to be separating. Our approach cannot detect such a type of wormholes, neither any other topological approaches.

VI. Light-Weight Approaches for Wormhole Detection

In previous sections, we characterize the topological impacts of the wormhole, and propose the fundamental wormhole detection approach. This method can detect all wormholes that are detectable in terms of topology, as discussed in Section IV. In order to maximize the detection capability, this fundamental topological method has its inherent complexity on protocol design and implementation. In this section, we consider how to strike a proper balance between sophisticated capabilities and ease of implementation. Based upon the understanding of the topological impacts of wormholes, we tailor our fundamental detection method and accordingly propose derivative light-weight topological detection approaches to suit different requirements.

To maximize the detection capability, topological wormhole detection methods have to attempt to explore some global

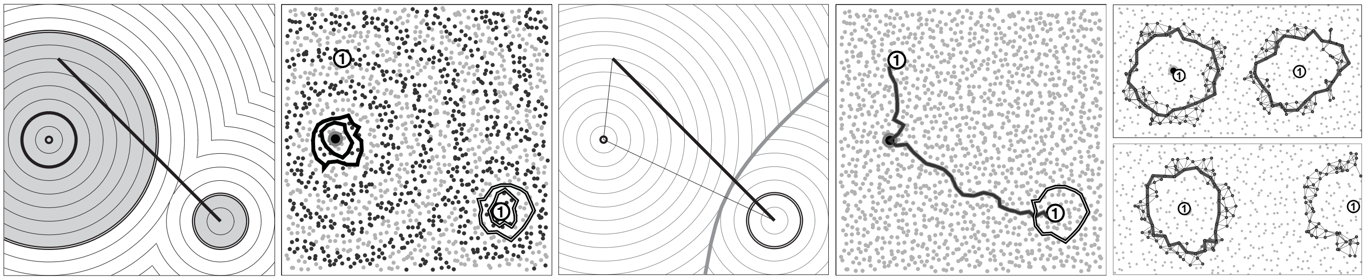

(a) (b) (c) (d) (e)

Fig. 9: Geometric and topological Wormcircle methods.

information, and have inherent complexity on protocol implementation. If we relax the objective of maximizing the detection capability of a topological method, i.e., leaving Class III and IV wormholes aside, we likely expect a low-complexity distributed or localized topological method with trade-off detection capability. We next customize our protocol to detect Class I and II wormholes efficiently with little global collaboration or only local operations.

A. Geometrical Wormcircle

We present the simplified method, called *geometrical Wormcircle*, focusing on detecting Class I and II wormholes. Similarly, we introduce the idea of Wormcircle in the continuous domain, and extend it into discrete networks. Consider an example shown in Figure 9 where one Class I wormhole resides on the surface. We select a random root point s, and run a continuous *Dijkstra* shortest path algorithm [29], as shown in Figure 9 (a). Each point thereafter is aware of its shortest geodesic paths to the root. Our main idea is to explore the structure of geodesic isolines, as the thin curves depicted in Figure 9 (a). Specifically, let $d(x, y)$ denote the geodesic distance between x and y on the surface S. A *ρ-level isoline* $I(s, \rho) = \{x \in S | \ d(x, s) = \rho\}$ is the set of points whose distances to root s are equal to ρ. The bold-line circle and double-line curves in Figure 9 (a) show two isolines of different levels. We can see that the double-line isoline contains two connected branches, an arc and an circle in the above example. We call the isoline circle around one wormhole endpoint *wormhole circle*. Wormhole circles are specific symptoms caused by wormholes. If we can detect the wormhole circles, we then locate the wormhole accurately.

Geometrical Wormcircle detects wormhole circles through exploring the geometrical characteristics of wormhole circles. We need to differentiate it from the legitimate isoline circles around the root point. The bold-line circle in Figure 9 (a) is a legitimate isoline circle that is not affected by wormhole in this example. For the legitimate d-level isoline circle C in the plane, its perimeter is $|C| = 2\pi d$. For the wormhole circle C_w with the same isoline distance d, however, its perimeter $|C_w| = 2\pi(d - d_0)$ is much smaller than the expected length $2\pi d$, where d_0 is the distance from the import endpoint of the wormhole to the root. We can validate and apply such an observation from continuous domain in discrete networks, i.e., to trace the perimeters and distance of the isoline circles from the root. We then obtain a distributed algorithm to detect the wormhole circles with connectivity information, as shown in Figure 9 (b). The procedures of tracing isoline circles are mostly similar with that of finding a skeleton circle in Section V-B. We skip the details here due to space limitations. The single-line and double-line loops in Figure 9 (b) depict the discovered discrete isoline circles. After detecting the isoline circle, we can estimate the perimeter of the circle and compares it with its isoline level to determine whether or not the detected isoline circle is a wormhole circle. As mentioned in the continuous case, for a legitimate d-level isoline circle C, the *perimeter-level ratio* $\gamma = |C|/d$ between perimeter $|C|$ and level d is 2π. In the discrete network, we qualify a legitimate circle by validating its perimeter-level ratio. The legitimate ratio is required to be greater than a threshold τ (in our most experiments, we find that setting τ to a constant slightly less than 2π, e.g. $\tau = 5 < 2\pi$, is a proper choice). Through testing perimeter-level ratio, we detect the wormhole circles denoted by double-line loops in Figure 9 (b). On the contrary, the bold single-line loops around the root node are legitimate because their perimeters are compliant to their isoline distance level.

B. Topological Wormcircle

Geometrical Wormcircle only needs to build one global shortest path tree, and thus reduce the complexity of fundamental method greatly. Geometrical Wormcircle, however, is not perfect because its effectiveness is influenced by the selection of the tree root. There are mainly two types of failure cases for geometrical Wormcircle. First, when the two endpoints of a wormhole are of nearly equal hop counts to the root node, there will be no wormhole circles formed around the wormhole ends. Second, when the outgoing end of the wormhole locates at the network boundaries (inner or outer), the wormhole circle is split by network boundaries and will not be detected by geometrical Wormcircle. An intuitive solution to handle such cases of geometrical Wormcircle is to launch it multiple times independently with multiple different root nodes. Thinking about this idea, we consider the extreme case that each node gathers its localized connectivity and build a localized tree rooted at itself, and propose the improved Wormcircle method, called *topological Wormcircle*, relaxing the dependence on the location of the root of the tree. As mentioned before, finding the proper wormhole symptom is the key to design a good countermeasure. Topological Wormcircle aims at making each node use only local connectivity information. The major challenges of topological Wormcircle design lie in how to explore the local impacts caused by the wormhole.

We first characterize the local topological features of wormholes in the continuous domain, and then explain its practical implementation in the discrete networks. Let S denote a plane region attached with one Class I or II wormhole link w. Let s be an arbitrary interior point in S. Recall that given a constant ρ, a ρ-level isoline around point s is $I(s, \rho) = \{x \in S | \ d(x, s) = \rho\}$, i.e., the set of points of distances to s equal to ρ. When ρ is small enough, supposing s is not located in one endpoint of the wormhole link w, the isoline $I(s, \rho)$ of s will contain only one connected branch, being a loop. Clearly, this loop is separating. We then analyze how the Class I and II wormholes affect the structures of an isoline locally. As a comparison, if s is located in one endpoint of the wormhole link w, it is not difficult to see that isoline $I(s, \rho)$ will contain more than one connected branches. Specifically, if wormhole link w is of Class I, $I(s, \rho)$ will comprise two cycles; if wormhole link w is of Class II, $I(s, \rho)$ is composed of two or more connected branches and one of them is a cycle. Based on the above observations, we use the local structures of an isoline of one point to determine whether or not there exists a wormhole within the neighborhood of the point.

To summarize above observations, if local isoline $I(s, \rho)$ of a point s contains at least one loop and has more than one connected branch, we can determine that there exists at least one wormhole endpoint within the region enveloped by $I(s, \rho)$. This is the key idea of the topological Wormcircle.

We can implement the principle of topological Wormcircle in the discrete wireless networks, as shown in Figure 9, where a Class I and Class II wormhole resides in the upper and lower network respectively. Considering the big-dot node s located at one endpoint of a Class I wormhole in the upper network shown in Figure 9 (e). Node s collects its k-hop neighboring connectivity, and locally computes the isoline around it. The local isoline of s contains two big connected components. Node s further can find one isoline loop from each component, denoted by the bold line. Similarly, we also can find two connected components for the Class II wormhole in the lower network when node s resides at one endpoint of a Class II. We here skip the details about the procedures of tracing isoline circles in discrete networks that have been discussed previously.

C. Remarks on Wormcircle Methods

The Wormcircle methods can be regarded as simplified instances of fundamental approach. We here focus on the case of geometrical Wormcircle method. The case of topological Wormcircle can be explained in the similar way.

Geometrical Wormcircle explores the geometrical characteristics of wormhole circles to detect wormholes. Suppose we cut along the double-line wormhole circle in Figure 9 (a). The surface preserves to be connected because the two filled gray regions are connected by the wormhole, while cutting along the legitimate bold-line isoline circle in Figure 9 (a) will divide the surface into two parts. The topological difference between legitimate isoline circles and wormhole circles is that a legitimate isoline circle is a separating loop while a wormhole circle is non-separating.

We use Figures 9 (c) and (d) to illustrate procedures of fundamental approach for the same example in Figures 9 (a) and (b). Fundamental approach shares the same first step of building a shortest path tree with geometrical Wormcircle. For convenience of intuitive understanding, we describe the further procedures of fundamental approach both in continuous domain and discrete network domain shown in Figures 9 (a) and (b), respectively. The light bold line shows the cut locus that contains only one cut path, and two thin lines and bold-line wormhole link compose a triangle, which forms a candidate loop. To detect Class I or II wormholes, the fundamental approach tries to seek a simple closed curve in the neighborhood of the candidate loop that crosses the candidate loop in odd times. These procedures validate that this candidate loop is an independent non-separating loop, and is thus associated with a degenerated genus introduced by the Class I or II wormholes. We therefore are ready to paraphrase geometrical Wormcircle in the language of fundamental approach, i.e., a wormhole circle is an non-separating loop that corresponds to the simple closed curve that is in the neighborhood of the candidate loop and crosses the candidate loop one time.

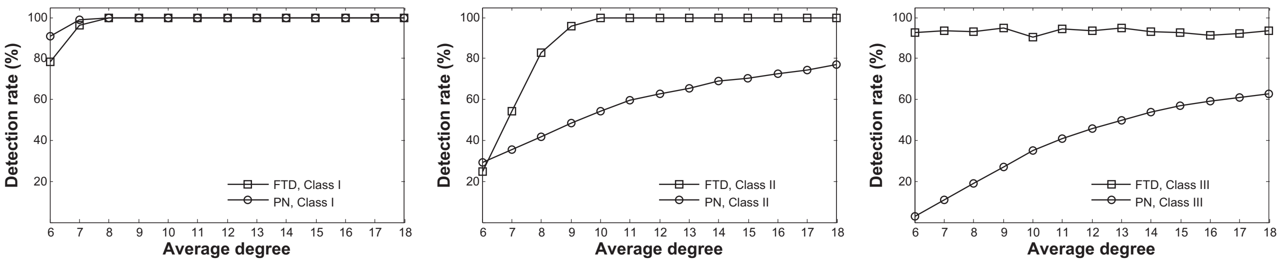
Wormcircle approaches focus detecting Class I or II wormholes in a light-weight manner and largely simplify the fundamental approach greatly. In particular, to detect all possible wormholes, previous fundamental method needs to perform further operations beside building the shortest path tree, including finding cut locus and candidate loops, testing the non-separating loops, and etc. Those further operations involve collaboration among the entire network, such as flooding different colors, which dominates the overall cost of computing and communication complexity. Wormcircle approaches cut down the construction of cut locus and candidate loops, and replace those further operations with light-weight operations, and thus simplify our algorithm dramatically while preserving the detection effectiveness for Class I and II wormholes. Hence, Wormcircle approaches make a beneficial trade-off between simplicity and effectiveness in topologically detecting wormholes.

VII. Evaluation

In this section, we examine the performance of this design in randomly generated networks and analyze the cost and security.

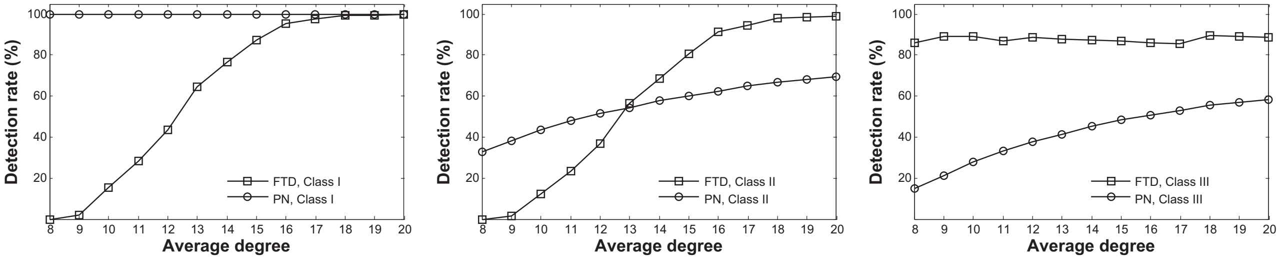
A. Performance Analysis

We conduct extensive simulations under various situations to evaluate the effectiveness of our approach. By varying node placement, node density, as well as the number and type of wormholes inside the network, we evaluate the rate of successfully detected wormholes. We compare our fundamental topology deviations based approach (denoted as FTD) with the packing number based approach (denoted as PN) proposed by Maheshwari et al. [12], which is to the best of our knowledge the only distributed method using solely node connectivity to detect wormholes.



(a) (b) (c)

Fig. 10: Detection rates against different node degrees and types of wormholes in the perturbed grid distribution



(a) (b) (c)

Fig. 11: Detection rates against different node degrees and types of wormholes in the random distribution.

1) Simulation Setup and Evaluation Approach:

In our simulations, the basic network layout is the same as the example shown in Figure 6, i.e., a 600m by 600m square area with multiple holes inside. We fill the area with a network of 3200 nodes and embed a single wormhole in the network. During our simulation we test our approach on various network fields of different shapes, and obtain consistent results. We omit presenting the results due to the space limitation. We evaluate the algorithms with parameters in three orthogonal dimensions, *node distribution* model, *network density*, and *wormhole classification*. By default, for each set of simulation, we conduct 100 runs with different node generations and report the average.

Node distribution model. In our simulations, nodes are deployed using two models: *random placement* and *perturbed grid*. In the random placement model, nodes are randomly deployed over the field, corresponding to an ad hoc organization of a network, e.g., dropping sensors from an airplane. Such a model contains inherent irregularities in the network topology. In the perturbed grid model, we deploy nodes on a grid and then perturb each node with a random shift. This model has been adopted [12, 32] to approximate manual deployments of nodes, corresponding more closely to planned organizations of a wireless network, e.g., organizing nodes in an indoor environment. Perturbed grid uniformly fills sensors into the field.

Network density. We evaluate the rate of successfully detected wormholes by varying the average node degree. We use basic UDG model to build the network. Note that our detection approach does not enforce the compliance to specific communication models for the network. Using UDG model is for the convenience of comparison with PN approach, which strictly relies on the UDG model. We vary the communication radius of sensors to yield average node degrees from 6 to 20.

Wormhole Classification. As mentioned before, different types of wormholes are of different difficulty to detect. We verify the effectiveness of our approaches about wormholes in varied classes. In each run, we randomly place a wormhole with different Classes inside the network with at least 8 hop span. More concretely, the nodes in the network locating near on the borderline of network deployment area are regarded as boundary nodes [32]. One wormhole endpoint is considered to locate at the boundary of the network, if this endpoint only allures nodes on the boundary. The wormhole endpoint is regarded to be inside the network if all nodes neighboring to this endpoint are distant to network boundary above k hops, e.g., 4 hops.

2) Analyzing the Results:

The results are displayed in Figures 10 and 11. Let us first consider the results in the network of perturbed grid model in Figure 10. For Class I wormholes both our approach and the packing number based approach can achieve nearly 100% detection rate even under low node density. For the cases of Class II and III wormholes, the packing number based approach bears relatively low detection rate, while our approach rapidly approaches 100% detection rate when the node degree rises above 9. This is mainly because in packing number based approach, the probability of the appearance of forbidden structures around Class II and III wormholes reduces dramatically when wormhole endpoints locate on network boundaries. Instead, our approach successfully captures the global impact of Class II and III wormholes by detecting non-separating loops (pairs). Further, an interesting behavior can be observed from Figure 10 (c). The detection rate of Class III in our approach is independent of the average node degree. This is due to that the partner loops in the detection of Class III wormholes is much longer than the locally contractible cycles in the case of Class I and II. These long cycles can still form even when the average degree is relatively low.

We then examine the results in the random deployment model and display the results in Figure 11. For both our approach and the packing number based approach, the random deployment provides slightly lower but still increased detection rates as the node density increases. For our method,

TABLE I: Message Complexity of Different Approaches

Approaches	FTD	GW	TW	PN
Message Complexity	$O(n^{3/2})$	$O(n)$	$O(n)$	$O(n)$

it approaches nearly 100% when the average node degree increases to 18 for both Class I and II wormholes. Generally, the performance in random node deployment is not as satisfactory as perturbed grid, due to more irregularities in the random deployment. When the node density is small (average node degree <12), it is difficult for one node to find discrete circles of sufficiently small sizes to verify the non-separating loops (pairs) because of the poor connectivity.

B. Cost Analysis

We analyze the message complexity of our fundamental approach, geometrical Wormcircle (GW), and topological Wormcircle (TW) approaches, and compare them with PN approach. Summary of those analytical results is shown in Table I. It is not difficult to obtain the $O(n)$ complexity for PN, TW and GW approaches, where n is the total number of nodes in the network.

We here mainly investigate the complexity of our fundamental method. We examine the message complexity in each step of this method. The first step involves building a shortest path tree, and has a message complexity of $O(n)$. The second step is to find cut pairs from the shortest path tree to build candidate loops. From continuous version of this algorithm as described in Section IV, we know that there are at most $N_{cut} = 6(g+d)+3b-3$ number of cut paths in a wormhole-infected surface $\tau(g,d,b)$ with genus g, degenerated genus d and boundaries b. We assume that the original surface $\tau_0 = \tau(0,0,b_0)$ has a constant b_0 number of boundaries and is mounted a constant N number of wormholes. We know $N = 2g+d+b-b_0$ from Theorem 2. The number N_{cut} of cut paths is then bounded in a constant less than $6(N+b_0)$. Correspondingly, the number of cut pairs in the discrete case is bounded in DN_{cut}, where D is the network diameter. From each cut pair, one candidate loop is constructed at the cost of message complexity $O(D)$, based on two conditions as discussed by Wang et al. [32], including checking the least common ancestor of the pair of nodes etc. The message complexity of building all candidate loops is $O(D^2)$. Each node in a candidate loop can utilize the localized connectivity to check Class I or II wormholes. Hence, the message complexity of detecting all Class I or II wormholes is $O(D^2)$. To detect Class III wormholes, our protocol needs flooding two colors from the two sides of each cut path to find knit non-separating loop pairs. The total message complexity in this step is $O(Dn)$. After we aggregate these message costs together and set the approximation order of network diameter D to be $\sqrt{n}$, we obtain the message complexity of our fundamental approach $O(n^{3/2})$.

C. Discussion of security issues

Having examined the performance and cost of our protocol, we discuss the security of this wormhole detection method and possible security enhancements for our detection protocol. This section mainly gives heuristic arguments and a rigorous analysis is left as future work.

During the execution of this protocol, data packets are generated and transmitted among nodes to detect wormholes. If data communication over each hop is secured, our protocol is secured accordingly. We mainly need to protect the data transmission over wormhole links. If the wormhole attacker performs packets tunneling reliably and honestly, data packets can be transmitted over wormholes as normal links. The adversary, however, can implement unauthorized malicious attacks on the packets passing wormholes, i.e. overhearing, corrupting, replaying, injecting, or dropping messages in the communication channel. In the following, we present specific security mechanisms to ensure the correctness of this protocol.

We first discuss the assumptions and security objectives. In wormhole attack model described in Section III-A, we adopt the Dolev-Yao attacker model and consider wormhole attacks as outsider attacks, the same as most previous works [4–12]. We assume the reasonable security level to outsider attacks, i.e., the network is provided with key management mechanisms to establish symmetric keys between each sender and its receivers, e.g., through key pre-distribution or agreement techniques during network bootstrap. The requirements (or objectives) of data security over wormhole links are basically the same as those well defined in the traditional networks, i.e., data confidentiality, authenticity, freshness and availability. More specifically, those requirements can be further elaborated in wormhole attacks as follows. Confidentiality means that we prevent adversaries from learning information about the messages passing wormholes. Authenticity and integrity require preventing unauthorized parties from participating in the network. Legitimate nodes are able to detect messages from the unauthorized attacker and reject them. If the adversary modifies a legitimate message passing wormholes, the receiver is able to detect such a tampering. Freshness ensures that the messages passing the wormholes are just-in-time instead of old messages replayed by the adversary. Availability means that the expected data communication over wormhole links can be successfully performed instead of being dropped.

We can employ the security mechanisms in both link layer and the upper layers to achieve those objectives in the above. For example, MiniSec [25] can efficiently achieve authenticated encryption with low energy and communication consumption, i.e. 1.1% overhead in energy and 8.3% overhead in communication, respectively. In MiniSec, data authentication is achieved by attaching the message authentication code (MAC) to a packet. The receiver uses shared secret key and nonce to recompute the MAC of the packet and matches it with the received MAC to validate the packet. Data confidentiality is provided by a cryptographic encryption scheme with a probabilistically unique initialization vector for each encryption. MiniSec provides both authentication and secrecy using block-cipher encryption mode with a non-repeating counter. Meanwhile, every node maintains a record of the last value from every sender it receives, it rejects packets with an equal or smaller counter value. Therefore, an attacker cannot replay any packet that the receiver has previously received. MiniSec is able to provide the three important properties of secure communication in the link layer: confidentiality, authenticity, and freshness. With such a setting, the adversary still can drop data packets in the link or physical layers. Such drops, however, are blind and they can be easily recovered by techniques of reliable routing [27], e.g. retransmission for unstable links, or be explicitly identified as unreliable links through reputation based schemes [14, 20, 34].

VIII. Conclusions

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, leading to low applicability. In this work, we fundamentally analyze the wormhole issue by topology methodology and by observing the inevitable topology deviations introduced by wormholes. We generalize the definition of wormholes, classify the wormholes according their impacts on the network and propose a topological approach. By detecting non-separating loops (pairs), our approach can detect and locate various wormholes and relies solely on topological information of the network. To the best of our knowledge, we make the first attempt towards a purely topological approach to detect wormholes distributedly without any rigorous requirements and assumptions. Our approach achieves superior performance and applicability with the least limitations.

References

[1] I. Aad, J.-P. Hubaux, and E. W. Knightly, "Impact of denial of service attacks on ad hoc networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 791–802, 2008.

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

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

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

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

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

[7] L. Lazos, R. Poovendran, C. Meadows, P. Syverson, and L. W. Chang, "Preventing wormhole attacks on wireless ad hoc net-works: A graph theoretic approach," in *Proc. of IEEE WCNC*, 2005.

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

[9] Y. Hu, A. Perrig, and D. Johnson, "Wormhole attacks in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 370–380, 2006.

[10] J. Eriksson, S. V. Krishnamurthy, and M. Faloutsos, "Truelink: A practical countermeasure to the wormhole attack in wireless networks," in *Proc. of IEEE ICNP*, 2006.

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

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

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

[14] 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.

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

[16] 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.

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

[18] L. Lazos and R. Poovendran, "HiRLoc: High-resolution robust localization for wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 233–246, 2006.

[19] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE JSAC*, vol. 24, pp. 247–260, 2006.

[20] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, "ODSBR: An on-demand secure Byzantine resilient routing protocol for wireless ad hoc networks," *ACM Transactions on Information and System Security*, vol. 10, no. 4, Article No. 6, 2008.

[21] B. Awerbuch, R. Curtmola, D. Holmer, H. Rubens, and C. Nita-Rotaru, "On the Survivability of Routing Protocols in Ad Hoc Wireless Networks," in *Proc. of IEEE SECURECOMM*, 2005.

[22] J. Eriksson, M. Faloutsos, and S. Krishnamurthy, "Routing amid colluding attackers," in *Proc. of IEEE ICNP*, 2007.

[23] C. Castelluccia, A. Francillon, D. Perito, and C. Soriente, "On the difficulty of software-based attestation of embedded devices," in *ACM CCS*, 2009.

[24] Y. Yang, X. Wang, S. Zhu, and G. Cao, "Distributed software-based attestation for node compromise detection in sensor networks," in *IEEE SRDS*, 2007.

[25] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "MiniSec: a secure sensor network communication architecture," in *ACM/IEEE IPSN*, 2007.

[26] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *Proc. of ACM SenSys*, 2004.

[27] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proc. of ACM SenSys*, 2003.

[28] A. Hatcher, *Algebraic Topology*. Cambridge University Press, 2002.

[29] J. Erickson and K. Whittlesey, "Greedy optimal homotopy and homology generators," in *Proc. of ACM-SIAM SODA*, 2005.

[30] J. Erickson and S. Har-Peled, "Optimally cutting a surface into a disk," in *Proc. of ACM SoCG*, 2002.

[31] M. J. Pelsmajer, M. Schaefer, and D. Stefankovic, "Removing even crossings, continued," in *DePaul CTI 06-016*, August 28 2006.

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

[33] R. Ghrist, D. Lipsky, S. Poduri, and G. Sukhatme, "Surrounding nodes in coordinate-free networks," in *Proc. of Workshop in Algorithmic Foundations of Robotics*, 2006.

[34] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *ACM MobiCom*, 2000.

Dezun Dong (S'09) received his B.S. and M.S. degrees at National University of Defense Technology (NUDT), China, in 2002 and 2004, respectively. He is currently a PhD student at School of Computer, NUDT, and visiting at the Department of Computer Science and Engineering, Hong Kong University of Science and Technology.

Mo Li (M'06) received the B.S. degree from Tsinghua University, Beijing, China, in 2004, and Ph.D. degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, in 2009. He is currently an Assistant Professor of School of Computer Engineering, Nanyang Technological University.

Yunhao Liu (M'02-SM'06) received the B.S. degree in automation from Tsinghua University, China, in 1995, and the M.S. and Ph.D. degrees in computer science and engineering from Michigan State University in 2003 and 2004, respectively. He is an Associate Professor of Department of Computer Science and Engineering at Hong Kong University of Science and Technology.

Xiang-Yang Li the received M.S. (2000) and Ph.D. (2001) degrees at Dept. of Computer Science from University of Illinois at Urbana-Champaign. He received the B.S. degree in Computer Science from Tsinghua University, China, in 1995. Currently he is an Associate Professor of Department of Computer Science, Illinois Institute of Technology.

Xiangke Liao received the B.S. and M.S. degree in computer science from Tsinghua University and National University of Defense Technology (NUDT), China, in 1985 and 1988, respectively. He is now a Professor and the Dean at School of Computer, NUDT, China.