Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2016, Article ID 3405264, 15 pages
http://dx.doi.org/10.1155/2016/3405264

Research Article

Wormhole Detection Based on Ordinal MDS Using RTT in Wireless Sensor Network

Saswati Mukherjee,[1] Matangini Chattopadhyay,[1] Samiran Chattopadhyay,[2] and Pragma Kar[1]

[1]*School of Education Technology, Jadavpur University, Kolkata, India*
[2]*Department of Information Technology, Jadavpur University, Kolkata, India*

Correspondence should be addressed to Saswati Mukherjee; saswatimuk@gmail.com

Received 7 August 2016; Revised 26 September 2016; Accepted 10 October 2016

Academic Editor: Gianluigi Ferrari

In wireless communication, wormhole attack is a crucial threat that deteriorates the normal functionality of the network. Invasion of wormholes destroys the network topology completely. However, most of the existing solutions require special hardware or synchronized clock or long processing time to defend against long path wormhole attacks. In this work, we propose a wormhole detection method using range-based topology comparison that exploits the local neighbourhood subgraph. The Round Trip Time (RTT) for each node pair is gathered to generate neighbour information. Then, the network is reconstructed by ordinal Multidimensional Scaling (MDS) followed by a suspicion phase that enlists the suspected wormholes based on the spatial reconstruction. Iterative computation of MDS helps to visualize the topology changes and can localize the potential wormholes. Finally, a verification phase is used to remove falsely accused nodes and identify real adversaries. The novelty of our algorithm is that it can detect both short path and long path wormhole links. Extensive simulations are executed to demonstrate the efficacy of our approach compared to existing ones.

1. Introduction

Security issues of a Wireless Sensor Network (WSN) are a significant concern since sensors are deployed in a hostile environment. Sensor nodes are vulnerable to both external and internal attacks. Wormhole attack is a type of external attack initiated by pairs of colluding attackers as shown in Figure 1. These pairs of colluding attackers are connected by low latency links, namely, wormhole links. In this paper, the phrase "wormhole links" is synonymously used as "wormhole tunnels". High frequency or wired links are used to establish these low latency links. At one end of the link the attacker captures the packets, tunnels them via wormhole link, and replays the packets at the other end once the link is established [1]. Thus, the distant sensor nodes around the two ends of wormhole links consider each other as neighbours although they are far from each other. Each wormhole node is capable of faking a route that is shorter than the original route. By building this high speed tunnel, a wormhole attack can disrupt the routing mechanism, attract a large amount of traffic, and also launch selective forwarding attack. Moreover, the wormhole links also exploit some sophisticated attacks like man-in-the-middle attack, cipher breaking attack, and denial of service attack.

Several solutions have been proposed to repulse wormhole attack in the literature. However, most of the techniques have their own limitations like requirements of synchronized clock [2], positioning device, or directional antenna [3], which increase the hardware cost of the system. Some existing solutions use neighbour mismatch method [1], Round Trip Time (RTT) calculation with message encryption using hash function [4], and topological comparison using new packet type [5]. But these solutions have certain limitations like wormholes remaining undetected in sparse network, increased message overhead, and so forth. In addition, some localization-based approaches are proposed to relax these limitations. However, most of them have some restrictions. For example, node labelling scheme requires neighbour

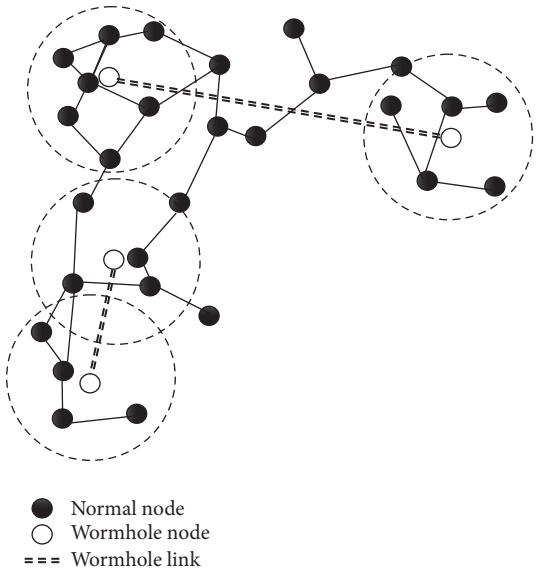


Figure 1: Wormhole attack scenario.

finding approach, which restricts its efficiency in large scale networks. In short, the severity of effects of wormhole attack in WSN, countermeasures, and their limitations motivate us to propose a novel efficient scheme to defend against wormhole attacks.

Wormhole tunnels can be of two types: long path and short path. In long path wormhole tunnel, the colluding pair of wormhole nodes is positioned far away from each other whose distance is equivalent to k hops where k is greater than a sufficiently large constant. Long path wormhole tunnels have diverse and significant effect on the network topology since it partitions the network into two or more subnetworks. On the other hand, in short path wormhole tunnels, the wormhole nodes at both ends of the tunnel are relatively shorter than 2 hops distance. Although short path wormhole tunnels do not partition the network, they increase packet dropping and, hence, network traffic is affected. Short path wormhole tunnels may have interferences or overlapping ranges which makes them even more difficult to be identified by normal range-based detection techniques.

Long path wormholes have been studied in [6–8]. In this paper, we have proposed algorithms for detection of long path as well as short path wormhole tunnels.

In this work, we propose a novel range-based topology comparison method for wormhole detection in WSN. Behaviour of wormhole attackers in the network disrupts the network connectivity; so, an abnormal network structure is introduced by the wormholes which needs to be explored. Each node gathers the RTT of a message and measures the distance from its neighbours. Based on this distance, a neighbourhood subgraph is generated. Then, by using shortest path algorithm, an estimation distance matrix is constructed between all node pairs. Next, we apply the ordinal Multidimensional Scaling (MDS) on the estimation distance matrix to restructure the subgraph and embed it on a plane. Ordinal MDS takes this estimation distance matrix as input and produces a spatial configuration of nodes by assigning virtual node positions (i.e., node coordinates). The estimated distances between the node pairs are ranked and are compared with the virtual distance for monotonicity relation. The disparity caused by the disagreement of the rank order is reduced via iterative computation of node coordinates and the corresponding virtual distance is updated until the virtual distance agrees with the rank order of the estimated shortest distance. The underlying observation of our wormhole detection is as follows. If i is a wormhole then a deviation can be observed between the virtual distance and the estimated distance even after iterative computation of virtual coordinates of the nodes. Otherwise, if i is a normal node, the virtual distance is a mere approximation of the estimated distance. The nodes which are violating the monotonic property are put to the suspect node set. Then, a filtering technique is used to determine the real wormhole link from the suspect node set.

It is well studied in [9] that the wormhole nodes hide the tunneling/propagation delay when radio transmission is used. So, this characteristic can be used to detect short path wormhole tunnel instead of relying solely on topological features of the network. Hence, we consider using the RTT of a message. We assume that the wormhole nodes connected by short path are present in the network if RTT of a message between two ends of a short wormhole path is abnormally less than the $1/k$ times of the average RTT of all normal links.

The main contributions of this work are as follows.

(i) We propose a wormhole detection algorithm that can defend both the long path and the short path wormhole links. For long path wormholes, there are almost no false positives but, for short path wormhole tunnels, some false positives do exist.

(ii) Our approach does not require any deterministic threshold to generate probable wormholes.

(iii) We adopt a method to eliminate the true nodes which are initially suspected.

(iv) Our proposed method has been compared with the state-of-the-art TRM method [1], WORMEROS [4], and MDS-based local connectivity [7]. Our method can detect all wormhole nodes with fewer false positives.

The rest of the paper is organized as follows. Section 2 presents the previous research efforts that contribute to our approach. In Section 3, the model definition of our work is introduced. Section 4 presents the detailed outline of our approach. Three principal modules of the proposed mechanism: efficient network reconstruction, suspicion phase, and verification phase, are described in detail. In Section 5, influence of several parameters and time complexity of our approach are discussed. Section 6 presents the experimental results obtained through simulation. Section 7 concludes the paper and discusses the future extensions.

2. Related Work

2.1. Wormhole Detection in Wireless Networks. There are several approaches for wormhole attack detection based on the symptoms introduced by wormhole links.

The first line of defence is based on special hardware devices that use geographical leashes and temporal leashes [2]. Leash is information added to the packet in order to defend wormhole attack. In geographical leashes, GPS technology is required to capture the node location included in the packets and all the nodes require loosely synchronized clock. Temporal leash ensures that each packet transmitted between source and destination nodes has packet expiration time which limits the maximum travel distance. All the nodes need to have tightly synchronized clock. However, to protect the leash, authentication data is added to the packet which increase the communication and processing overhead. In [3], directional antennas are used to locate the infeasible communication; however, it mitigates wormhole problems partially and degrades network connectivity by rejecting legitimate neighbours. Čapkun and others [10] proposed mutual authentication with distance-bounding algorithm to detect wormhole attacks without requiring any clock synchronization. But this approach uses special hardware for accurate time measurements. All these countermeasures increase the hardware cost of the system and are impractical in resource constraint devices.

The second line of defence relies on RTT of the packets and topology comparison [5]. Clock synchronization is an issue for distance estimation between source and destination for every packet but topology comparison eliminates this limitation. However, the scheme in [5] uses special packet type which increases the communication overhead. WORMEROS [4] detects and eliminates wormholes in two consecutive phases. Firstly, RTT-based neighbour finding method is applied without intermediate node cooperation. A deterministic threshold value is used to compare the RTT values between the source and destination and RTT values higher than the threshold is considered as suspect wormhole link. Secondly, encrypted message exchange with different frequency confirms the wormholes. This algorithm detects all wormholes but has more false positives. In [11], authors gather the hop count value and delay associated with the disjoint paths and calculate the propagation delay per hop to detect wormhole attacks. It considers that the delay per hop of the path under attack is much larger than the normal path. Flooding mechanism is used to calculate and compare the delay per hop.

The third line of defence is based on neighbour mismatch which exploits the local neighbourhood information based on expanding the transmission range [1]. Although the scheme is lightweight and requires no additional hardware, the algorithm efficiency drops in case of sparse network as the entire method depends on neighbour set comparison. In [12], the RTT between two successive nodes is considered for wormhole detection and the neighbour set of each node is compared with other successive nodes. The scheme does not require any specific hardware but it suffers from increased memory overhead since each node needs to store the neighbour list. LiteWorp [13] introduces a notion called guard nodes which overhear the transmission of two neighbouring nodes and determines the malicious behaviour of one of its neighbour. However, in sparse network, finding a guard node is not always feasible for a particular link. MobiWorp [14] is an extension of LiteWorp which assumes availability of location information. Choi and others [15] use neighbour node monitoring of each node and detect fake neighbours that are beyond the transmission range.

The fourth line of defence uses graph mismatch under connectivity models. Poovendran and Lazos [16] present a location-based solution by using guard nodes that have extra communication range to deal with wormhole attacks. Establishing multihop pairwise keys is not properly handled in the said graph-based framework.

The fifth line of defence considers statistical mismatch in traffic flow to detect the presence of wormhole links. In split multipath routing (SMR) protocol [17], a statistical analysis and time constraint algorithm rely on a drastic deviation in the routing statistics stored in the sink node under wormhole attack. It is unable to detect multiple wormholes in the network. PWORM [18] introduces packet marking scheme to gather routing information against packet drop. It calculates the frequency of a node appearing in the path because wormhole attack attracts more traffic and appears more frequently than normal nodes. Path length variations are calculated to localize the wormhole.

In the last line of defence, abnormality in the network topology due to wormhole tunnels is studied. In [8], wormhole attack is detected by identifying some prohibited substructures in the network connectivity that are not generally observed in normal connectivity graph. This method works well in Unit Disk Graph (UDG) connectivity model but is inaccurate for non-UDG model. In [19], authors propose MDS visualization of wormholes (MDS-VOW) that detects wormholes by visualizing anomalies in the reconstructed network formed by using the MDS technique. This scheme uses a centralized approach and it is suitable if one malicious node resides at both ends of the wormhole links. For large scale network, Wang and Lu [9] enhance the MDS-VOW and propose an interactive wormhole detection method, which monitors the topology changes based on real time visualization approach known as interactive visualization of wormholes (IVoW). However, this detection requires domain knowledge and expertise to solve visual analysis problems. Another connectivity-based approach is proposed in [6] that uses bipartite subgraph theory to remove wormholes. The algorithm is robust in different communication models but suffers from many false positives. Dong and others [20] propose a wormhole detection method that relies on network connectivity information. It detects wormholes in a distributed manner by observing topology deviations. It is suitable for both continuous and discrete geometric terrain where each node maintains connectivity with neighbouring nodes on the surface. However, increased node density is an issue in the detection performance. Chen and others. [21] propose DV-hop localization scheme which calculates hop count between the anchor nodes and estimates the average size for one hop. Location of each unknown node is estimated by using maximum likelihood estimation. In [22], mobile beacon and positioning scheme are used for wormhole detection. The mobile beacon can localize the attacker by estimating the center of the attackers' communication area. In [7], MDS is executed to reconstruct the local subgraph of each

Table 1: Summary of different techniques to detect wormholes.

Defense	Basic method	Pros	Cons
Wu et al. [1]	Local neighbourhood information	Lightweight	Less efficient in sparse networks
Hu et al. [2]	Leashes	Authentication protocol using symmetric cryptography	Communication and processing overhead
Hu and Evans [3]	Cooperative protocol using directional antennas	Uses one hop neighbour information	Detects partial wormholes and degrades network connectivity
Vu et al. [4]	RTT calculation and encrypted message exchange	High detection rate	High False positives and computation overhead
Alam and Chan [5]	RTT calculation and topology comparison	Topology comparison reduces false positives	Communication overhead
Ban et al. [6]	Bipartite subgraph theory	Robust in different communication models	High false positives
Lu et al. [7]	Connectivity-based approach	Less false positives	Short path wormholes remain undetected
Maheshwari et al. [8]	Graph connectivity	Efficient in UDG model	Inaccurate in non-UDG models
Wang and Lu [9]	Interactive visualization of wormholes (IVoW)	Improves detection efficiency	Requires domain knowledge and expertise to solve visual analysis problems
Čapkun et al. [10]	Distance-bounding algorithm	No additional clock synchronisation	Special hardware needed for time measurement
Chiu and Lui [11]	Hop count and delay calculation	High detection rate	Memory overhead
Tun and Maw [12]	RTT and number of neighbours calculation	No hardware required	Memory overhead
Khalil et al. [13]	Local monitoring using guard nodes	Lightweight and suitable for resource constraint networks	Not efficient in sparse networks
Khalil et al. [14]	Isolate attackers by secure central authority (CA)	Isolate attackers with increased scalability; low detection latency	Increased message exchange between CA and mobile nodes
Choi et al. [15]	Neighbour node monitoring	Timer prevents wormhole attacks without requiring clock synchronization	Does not support DSR optimization
Poovendran and Lazos [16]	Location-based and decentralized	Time synchronization not required	Packet transmission overhead
Zhao et al. [17]	Statistical analysis of routing detect wormholes	Lightweight	Detection rate declines in presence of multiple wormholes
Lu et al. [18]	Real time secure packet marking algorithm	Detects both active and passive attacks	Wormholes remain undetected in less traffic scenario
Wang and Bhargava [19]	MDS-visualization of wormholes	Efficient in case of single wormhole; less false positives	Centralized approach
Dong et al. [20]	Distributed approach using network connectivity information	Suitable for contiguous and discrete geometric terrain	Increased node density affects detection performance
Chen et al. [21]	DV-hop localization	Range-free localization	Intolerant to packet loss
Chen et al. [22]	Mobile beacon and positioning scheme	Energy efficient and high detection probability	Require GPS enabled beacon node to detect wormholes

node based on its neighbour information. The reconstructed network is then validated and verified for detecting probable wormhole nodes. A refinement technique is used to exclude suspect nodes and remove false positives.

Different techniques of wormhole detection along with their pros and cons are summarized in Table 1.

In this work, we propose a novel wormhole detection mechanism with almost no false positives and improved detection performance in both sparse and dense networks.

2.2. Ordinal MDS and Its Applications. MDS is a collection of techniques which embed dissimilar data for a given dissimilarity matrix in a selected dimension space. The embedding is often used to visualize and analyze exploratory data [23]. MDS is applied on the dissimilarity matrix which produces a position of objects in a small dimensional space as output. The basic objective of MDS is to find the coordinates of objects in a p dimensional space so that there is a good agreement between the observed dissimilarity and the

interobject distances. Traditional MDS techniques are categorised into metric and nonmetric MDS. In metric MDS, the dissimilarities between objects are linear to Euclidean distances whereas, in nonmetric or ordinal MDS, the dissimilarities exhibit a monotonic transformation with the Euclidean distances. Ordinal MDS nevertheless finds a good embedding in Euclidean space by monotonic regression. It gains better performance since it demands less rigid relationship between dissimilarities and distances. For the last few years ordinal MDS has widely been applied for node localization issues in WSN [24–28]. Miao and others [25] propose a RI-MDS localization algorithm that combines metric and nonmetric MDS and use affine transformation to translate relative coordinates to absolute ones. Nhat and others [24] propose NMDS-TOA localization algorithm that combines TDOA and MDS-map and uses sufficient number of anchor nodes to form the final estimated map. The significance of applying ordinal MDS over metric MDS is as follows.

(1) Ordinal MDS gains better performance by matching the disparity as closely as possible with the virtual distance having the restriction that the disparity maintains a monotonic relationship with the estimated distances between node pairs.

(2) The scaling is based on the rank order of the disparity. Ideally, the ranks of the nodes based on estimated distance are monotonic with their ranks based on true Euclidean distance.

(3) Ordinal MDS computes the node coordinates iteratively and updates the virtual distance to improve rank order agreement between estimated distances and virtual distances of node pairs.

(4) Ordinal MDS compensates the distortion caused by distance measurement errors via iterations.

In our approach, we have used ordinal MDS to reconstruct the local subgraph of each node.

3. Model Definition

In this section, we define the proposed model along with reasons for choosing this model. For the purpose of wormhole detection, we have considered two types of nodes: normal and wormhole.

3.1. Network Model. The network model is structured with N sensor nodes deployed in a planar region and is denoted by a communication graph $G = (V, E)$. In this graph G, vertices V denote the nodes and edges E denote the communication bidirectional links. We have considered UDG as the connectivity model and the deployment environment is assumed to be random. The sensor nodes do not require any special hardware or globally synchronized clock. Moreover, nodes are considered to be static and initially none of the nodes is compromised. Initially, the transmission range of every node is assumed to be identical, that is, r, since each node is modelled as a UDG. But for the purpose of justifying our wormhole detection method, we have expanded the transmission range of each normal node to $R = 2r$, which means that each node is capable of collecting its neighbour information when $d \leq 2r$. Consider a node i in graph G; the neighbour of node i is denoted as $N_G^{2r}(i)$. Since our work is primarily range-based, we measure distance based on RTT denoted as Ω of a message between each node i and its neighbours $N_G^{2r}(i)$. We use $\Omega_G^{2r}(i, j)$ to denote the RTT between the nodes i and j in the network G.

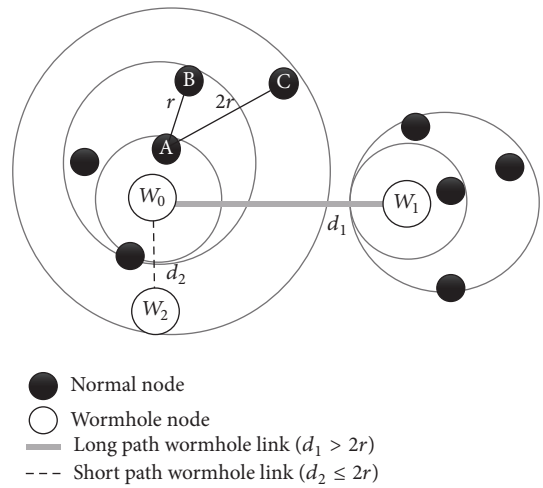


Figure 2: Illustration of proposed model.

3.2. Adversary Model

3.2.1. Wormhole Definition. Under wormhole attack, the malicious nodes generally work in pairs and establish a high speed, long distance tunnel between them. This tunnel or high frequency links create an illusion to the sensor nodes around the two ends of the link as direct neighbours. Wormhole nodes advertise a false short route to a destination, capture packet from one location, and transmit them to its paired wormhole node through the high speed tunnel.

In this work, we assume that the attacker can launch a wormhole attack without modifying any packet or compromising any node. Moreover, the cryptographic mechanisms and encryption keys that are shared between the nodes for secure communication remain unaware of the attacker.

Figure 2 depicts our model. In this figure, d_1 and d_2 represent the length of long path and short path wormhole links, respectively. In the network G, the path of the wormhole link e is assumed to be either long or short. In the long wormhole link, the wormhole peer(s) are placed far apart, so that the communication regions of the two ends do not overlap with each other. Nodes at the two ends of an edge e are denoted by $i(e)$ and $j(e)$. The shortest distance between $i(e)$ and $j(e)$ is denoted by $d_G(i(e), j(e))$. We have assumed that long path wormhole tunnel is present in the network if $d_G(i(e), j(e)) > 2r$, where $2r$ denotes the maximum transmission range of normal nodes. In addition, we have further assumed that wormhole attack prevails in the network even when the length of the wormhole tunnel is short; that is, the distance between the wormhole nodes is $d_G(i(e), j(e)) \leq 2r$.

4. RTT-Based Iterative MDS for Wormhole Detection

In this section, we put forward the detailed design and analysis of our wormhole detection approach based on ordinal MDS.

4.1. Outline. For the detection of the wormhole attack, range-based distance estimation and topological comparison are used to identify the wormhole links. The overview of our scheme is as follows. In the network, each node i gathers the RTT value of a message from its direct neighbours that are within the transmission range $2r$. In other words, a link is established between the node pairs if their distance is $d \leq 2r$. Based on the RTT measure, distances between node i and its direct neighbours are estimated and stored in a sparse matrix. This sparse matrix is used to construct the shortest distance matrix between all node pairs. Then, on this internode shortest distance matrix, ordinal MDS is applied which generates virtual node positions. Considering this estimated node position, Euclidean distance between the node pairs is calculated which generates virtual distance matrix.

However, the presence of wormhole link causes a disparity between estimated shortest path matrix and the reconstructed distance matrix. Ordinal MDS reduces this disparity via iterative computation of node position (i.e., node coordinates) and thereafter updating the Euclidean distance between the node pair. Then, the monotonic property is checked in the updated Euclidean distance matrix in relation to the RTT-based shortest distance matrix. Kruskal's Stress1 [29] measure is applied to check whether the updated node position fits the estimated dissimilarities, that is, the shortest path matrix. But, due to wormhole placement in the network, distance measurement errors cannot be compensated by mere approximation. On the other hand, absence of wormhole nodes results in a reconstruction which is a mere approximation of the estimated shortest distance matrix. The disparity between the reconstructed distance matrix and the estimated shortest path matrix affects the network reconstruction. The validity of the network reconstruction is very crucial in identifying the adversaries. In our approach, ordinal MDS is executed on each node to reconstruct the network and detect the probable wormholes on the basis of distortion in the reconstruction. The disparity generated via iterative MDS produces some false alarm and so filtering mechanism is adopted to remove the falsely accused nodes and identify the real wormhole links.

In the light of the above discussion, our detection approach mainly involves two modules: (a) applying ordinal MDS on shortest distance matrix for network reconstruction and (b) executing filtering techniques to identify the real adversaries. In the first module, the suspect node list is generated. The verification phase in the second module filters the normal nodes from the suspect list and final wormhole nodes are presented.

In the following section, these two modules are described in detail.

Input: A network graph $G(V, E)$
Output: Shortest distance matrix/Dissimilarity matrix $[P^{\Omega}_{i,j}]$
(1) **for** each node $i \in V$ **do**
(2) Each node i collects RTT values from its neighbours whose $d \leq 2$ and forms sparse matrix
(3) Construct neighbourhood subgraph $\Gamma^{\Omega}_{G}(i)$ of node i
(4) Apply ordinal MDS on $[P^{\Omega}_{i,j}]$ to reconstruct the subgraph
(5) **end for**

Algorithm 1: Floyd-Warshall shortest distance algorithm for shortest distance matrix.

4.2. Ordinal RTT-Based Reconstruction. In this section, the first module is divided into three phases for better understanding.

4.2.1. RTT-Based Distance Estimation. Two nodes are considered to be neighbours if $d \leq 2r$. Each node i in the network G collects the RTT, Ω, of a message from its neighbours and estimates the distance based on the travel time. We measure RTT, Ω, by sending ICMP PING packets from node i to its neighbour and receiving an acknowledgment back for the same packet. Theoretically, in a wireless environment, the RTT, Ω, of a message can be related to the distance d between nodes assuming that the wireless signal moves at a speed of light c. So we calculate the distance by the following formula:

$$d = c \times \frac{(\Omega)}{2}. \tag{1}$$

This measured distance d between every pair of sensor nodes that can hear each other is stored in a sparse matrix. Thus, each node i generates a neighbourhood subgraph denoted by $\Gamma^{\Omega}_{G}(i)$. Then, Floyd-Warshall shortest path algorithm is applied to calculate the shortest distance between all node pairs in $\Gamma^{\Omega}_{G}(i)$. The shortest distance matrix is denoted as $[P^{\Omega}_{i,j}]$. Algorithm 1 presents the Floyd-Warshall shortest path algorithm.

4.2.2. Network Reconstruction. In this section, nonmetric MDS technique is applied on the shortest distance matrix $[P^{\Omega}_{i,j}]$ which is also known as dissimilarity matrix to rebuild a network $\overline{\Gamma}^{\Omega}_{G}(i)$. In this network $\overline{\Gamma}^{\Omega}_{G}(i)$, a virtual position (i.e., node coordinates) is assigned to each node to calculate the Euclidean distance between each node pair (i, j) in $\overline{\Gamma}^{\Omega}_{G}(i)$. Thus a virtual distance matrix $[D^{m}_{i,j}]$ is generated. Each value in the matrix $[D^{m}_{i,j}]$ is denoted as $D^{m}_{i,j}$.

For ease of understanding, we subdivide the steps of ordinal MDS as follows.

(a) In nonmetric MDS, the disparity between each node pairs is related to Euclidean distance by some monotone function $d_{i,j} = f(\delta_{i,j})$. That is, in nonmetric MDS or ordinal MDS, for two pairs of nodes (i, j) and (k, l), if the shortest path distance of node pairs (i, j) is less than (k, l), then the

Euclidean distance of node pairs (i, j) is also less than (k, l) and vice versa.

Let (i, j) and (k, l) denote the two node pairs in network G, respectively. The distance between all the node pairs in the shortest distance matrix $[P^{\Omega}_{i,j}]$ is ranked from the smallest to the largest. The shortest path distances between two node pairs (i, j) and (k, l) are compared with the virtual distances of the corresponding node pairs in Euclidean distance matrix in $[D^m_{i,j}]$ to check the monotonic relationship. If there is any disparity in the distances of the two node pairs (i, j) and (k, l), that is, if $P^{\Omega}_{i,j} < P^{\Omega}_{k,l}$ and $D^m_{i,j} > D^m_{k,l}$, then Pool Adjacent Violators (PAV) algorithm [30] is applied to obtain a distance estimator $\overline{D}^m_{i,j}$. PAV algorithm works by averaging the distance of node pairs $D^m_{i,j}$ and $D^m_{k,l}$ which has violated the monotonic property with the distance of the preceding nonviolator. It is expressed by the following condition:

If $(P^{\Omega}_{i,j} < P^{\Omega}_{k,l})$ and $(D^m_{i,j} > D^m_{k,l})$, then

$$\overline{D}^m_{i,j} = \overline{D}^m_{k,l} = \frac{\left(D^m_{i,j} + D^m_{k,l}\right)}{2}. \tag{2}$$

Otherwise,

$$\begin{aligned} \overline{D}^m_{i,j} &= D^m_{i,j} \\ \overline{D}^m_{k,l} &= D^m_{k,l}. \end{aligned} \tag{3}$$

(b) Based on this average distance estimator, new coordinate x^m_i for each node $i \in G$ is computed by the following formula:

$$x^m_i = x^{m-1}_i + \frac{a}{n-1} \sum_{i \neq j} \left(1 - \frac{\overline{D}^{m-1}_{i,j}}{D^{m-1}_{i,j}}\right)\left(x^{m-1}_j - x^{m-1}_i\right), \tag{4}$$

where n = number of nodes, m = iteration counter, and a = iteration increment of steepest descent method.

Thereafter, the Euclidean distance $D^m_{i,j}$ is updated in the virtual distance matrix $[D^m_{i,j}]$.

(c) Then, Kruskal's Stress1 [29] measure is applied to test how well the reconstructed distance matrix or the new spatial node configuration fits the shortest distance matrix. In particular, value of Stress1 decreases if rank order agreement improves between shortest distance matrix and the reconstructed distance matrix. Ordinal MDS always aims to minimize the stress. Stress1 measure is denoted by the following formula:

$$\text{Stress1} = \sqrt{\frac{\sum^n_{i=0, j=0, i \neq j} \left(\overline{D}_{i,j} - D_{i,j}\right)^2}{\sum^n_{i=0, j=0, i \neq j} \left(D_{i,j}\right)^2}}. \tag{5}$$

The Stress1 measure is executed until the stress value is satisfied. That is, Stress1 < threshold ε. This threshold, ε, is obtained by observing how well the new configuration of nodes matches the shortest distance matrix, such that ε becomes constant after iterative computation of nodes' coordinates. If Stress1 > ε, the process is repeated; otherwise, it is terminated. The steps of ordinal MDS are presented in Algorithm 2 which generates the suspect node set.

4.2.3. Suspicion Phase. In this phase, a mechanism has been adopted to find which nodes can be suspected or challenged. We have observed in our simulation that after repeated iterations, the threshold, ε, becomes almost a constant value of 0.17. We are considering those nodes as suspect wormhole nodes which violate the monotonic property even after Stress1 < ε. We add those nodes in the suspect node set. As discussed in the adversary model, wormhole nodes introduced disparity between the shortest distance and the reconstructed distance. Since wormhole nodes are placed far apart, there is a significant mismatch between the shortest path distance and the virtual distance. However, there are some normal nodes which show such type of disparity too after the network reconstruction. After implementing this phase, all the suspect wormhole nodes are produced.

4.3. Verification Phase. In suspicion phase, some normal nodes may be mistakenly identified as suspect wormhole nodes which introduce false positive results. Removal of too many false positive nodes breaks the normal links and disrupts the network functionality. So, we use a filtering technique to verify and confirm real wormhole nodes and remove nodes involved in false positive.

4.3.1. Long Path Wormhole Link Detection. The wormhole nodes create a local structure; so, for detection of long path wormhole links, we use the theory of complete bipartite graph. Let W_1 and W_2 be two sets that contain wormhole nodes w_1 and w_2, respectively, in network G. Let the edge set be denoted by $W_1 \times W_2$ between the node pair $w_1 \in W_1$ and $w_2 \in W_2$. Considering the node sets W_1 and W_2 that contain nodes at the two ends of the wormhole, each node i is given the illusion that all the nodes in set W_2 are its direct neighbours. Thus, all edge sets $W_1 \times W_2$ share a node from both the set W_1 and the set W_2 and there are no edges formed between two nodes in the same set either in W_1 or in W_2. Thus, a complete bipartite subgraph G' of G is constructed.

Under wormhole-free environment, for each node pair $(w_1 \in W_1, w_2 \in W_2)$, the shortest distance is $d_{G'}(w_1(e), w_2(e)) \leq 2r$. Therefore, for any node pair (w_1, w_2), either there exists an edge e between w_1 and w_2 or there is a common neighbour between them. This happens iff $N^{2r}_{G'}(w_1) \cap N^{2r}_{G'}(w_2) \neq \phi$.

By carefully studying the behaviour of long path wormhole links, we can guarantee that there is no common element or node between the two ends of a wormhole tunnel. Thus, we arrive at Theorem 1.

Theorem 1. *There exists a long path wormhole between the node pair* $(w_1 \in W_1,\ w_2 \in W_2)$ *if the distance between them is much greater than twice the transmission range, that is,* $d_{G'}(w_1(e), w_2(e)) > 2r$, *and there is no common neighbour between* w_1 *and* w_2 *which is denoted by* $N^{2r}_{G'}(w_1) \cap N^{2r}_{G'}(w_2) = \phi$.

Theorem 1 is used to filter the suspect wormhole nodes using complete bipartite subsets. Firstly, all the connected components in this suspect nodes set S are identified. Let N be the set of such connected components. For detection of a wormhole pair, we consider only the connected component

Input: Shortest distance matrix/Dissimilarity matrix $[P^{\Omega}_{i,j}]$
Output: Suspect node set S
(1) Each node $i \in V$ is assigned an arbitrary initial location as (x_i, y_i)
(2) Set the threshold value $\varepsilon = 0.17$ for Stress1 measure
(3) Set iteration counter $m = 0$
(4) Compute Euclidean distance between each node pair $(i, j) \in V$ to generate virtual distance matrix $[D^m_{i,j}]$
(5) Apply monotone regression using PAV algorithm on $[P^{\Omega}_{i,j}]$ and $[D^m_{i,j}]$ to calculate disparity and get intermediate matrix $[\overline{D}^m_{i,j}]$
(6) **if** $(P^{\Omega}_{i,j} < P^{\Omega}_{k,l})$ and $(D^m_{i,j} > D^m_{k,l})$ **then**
(7) $\overline{D}^m_{i,j} = \overline{D}^m_{k,l} = (D^m_{i,j} + D^m_{k,l})/2$
(8) Add node i to the suspect node set S
(9) **else**
(10) $\overline{D}^m_{i,j} = D^m_{i,j}$
(11) $\overline{D}^m_{k,l} = D^m_{k,l}$
(12) **end if**
(13) Calculate new coordinate (x^m_i, y^m_i) for each node i_m using steepest descent method
(14) Set $a = 0.2$ as proposed by Kruskal [29]
(15) $x^m_i = x^{m-1}_i + (a/(n-1)) \sum_{i \neq j} (1 - \overline{D}^{m-1}_{i,j}/D^{m-1}_{i,j})(x^{m-1}_j - x^{m-1}_i)$
(16) $y^m_i = y^{m-1}_i + (a/(n-1)) \sum_{i \neq j} (1 - \overline{D}^{m-1}_{i,j}/D^{m-1}_{i,j})(y^{m-1}_j - y^{m-1}_i)$
(17) Recalculate the Euclidean distance of each node pair $D^m_{i,j}$ to update distance matrix $[D^m_{i,j}]$
(18) Calculate the Kruskal's Stress1 measure
(19) **if** Stress1 $> \varepsilon$ **then**
(20) Set $m = m + 1$
(21) Go to step (5)
(22) **else**
(23) Terminate
(24) **end if**

Algorithm 2: Steps of ordinal MDS algorithm for suspect node set.

and, hence, exclude isolated nodes. Then, we apply complete bipartite subgraph algorithm [31] for each connected component $C \in N$. The algorithm in [31] generates several complete bipartite subgraphs denoted as $(c_0, c_1, c_2, \ldots, c_x \in N)$. Let each node pair be denoted as $(w_1, w_2) \in C$ such that $w_1 \in W_1$ and $w_2 \in W_2$, where W_1 and W_2 are the two partitions of the bipartite subgraph. On each bipartite subgraph, we test the condition given in Theorem 1. For each C, we check $d_{G'}(w_1(e), w_2(e)) > 2r$. If $c_0 \in N$ is a complete bipartite subgraph that satisfies the condition $N^{2r}_{G'}(w_1) \cap N^{2r}_{G'}(w_2) = \phi$ then we have detected all the wormhole nodes connected by long path. Moreover, the basic aim of detecting the wormhole attack is to nullify them without disrupting the network functions. Since the wormhole nodes attract large volume of traffic, it is necessary to discard the increased network traffic passing through the wormhole link while retaining the functionalities of the nodes. We do this by removing the edges $W_0 \times W_1$ in the complete bipartite subgraph. The detection of long path wormholes is presented in Algorithm 3.

Input: Suspect node set S
Output: Long path wormhole nodes
(1) Identify all connected components N from S
(2) **for** each $C \in N$ **do**
(3) Find Complete Bipartite Set W_1, W_2 for node pair $(w_1, w_2) \in C$ such that $w_1 \in W_1$ and $w_2 \in W_2$
(4) **for** each $w_1 \in W_1$ and $w_2 \in W_2$ **do**
(5) **if** $d_{G'}(w_1(e), w_2(e)) > 2r$ **then**
(6) **if** $N^{2r}_{G'}(w_1) \cap N^{2r}_{G'}(w_2) = \phi$ **then**
(7) Discard the edges $W_1 \times W_2$
(8) **end if**
(9) **end if**
(10) **end for**
(11) **end for**

Algorithm 3: Long path wormhole detection algorithm.

4.3.2. Short Path Wormhole Link Detection. When two wormholes at two ends are very close to each other, those wormhole nodes are connected by short path. Thus, the shortest distance between the wormhole peer connected by short path is $d_G(w_1(e), w_2(e)) \leq 2r$. Eventually, these nodes get filtered out when wormholes connected by long path are detected.

So, merely relying on RTT-based distance estimation for detecting short path wormholes is not the right approach. We use RTT for detecting wormhole nodes connected by short path. The RTT between two ends of a short path wormhole tunnel is much less than the RTT between normal nodes since the wormhole nodes hide the propagation delay in radio transmission.

Input: Suspect node set S
Output: Short path wormhole nodes
(1) Each node i calculate RTT of each link
(2) Node i calculates average RTT of all its neighbours $\chi_G^{N(i)}$
(3) **if** $\Omega_G^{2r}(w_1', w_2') \leq \chi_G^{N(i)}/2$ **then**
(4) Confirm the link $\Omega_G^{2r}(w_1', w_2')$ as wormhole links connected by a short path
(5) **end if**

Algorithm 4: Short path wormhole detection algorithm.

According to Theorem 1, all possible node pairs are identified from the suspect list by applying the algorithm of complete bipartite subgraph. Let (w_1', w_2') be one such connected node pair that belongs to C (set of complete bipartite subgraphs) whose $d \leq 2r$. As mentioned in the adversary model, RTT between the wormholes connected by short path is abnormally less than average RTT of all normal links. Thus, the RTT between short path wormhole links is at most $1/k$ times of average RTT of all normal links as shown in Algorithm 4. In the following section, we analyze this condition.

5. Discussion on Parameters

In this section, we discuss how different parameters influence the performance of the detection method. Moreover, we also justify the value chosen for RTT.

Effect of r. In our experimental set up, we set $r = 2$. There are primarily two reasons. Typical length of a long path wormhole tunnel is set to 6 to 8 hops. The value of r is chosen so that $2r$ is just more than the length of the wormhole tunnel. If the value of r is increased such that $2r$ is much greater than the length of the wormhole tunnel, a larger neighbourhood subgraph will be generated due to which number of false positives will increase.

Effect of ε. The computation of Stress1 measure determines how smoothly the reconstructed network can be embedded on a plane. Setting the optimum threshold of Stress1 measure helps in identifying the suspect nodes in our approach. In particular, setting lower threshold fails to capture all wormhole nodes although it reduces the overhead of the verification phase. On the other hand, higher threshold captures all wormhole nodes but may introduce false positives and may increase overhead of the verification phase. It has been observed experimentally that $\varepsilon > 0.2$ induces false positives. So, $\varepsilon = 0.17$ is considered as optimum value where all the suspect wormhole nodes are included in the suspect set for further filtering that results in few false positives.

Effect of k. For short path wormhole tunnel detection, let the RTT between (w_1', w_2') be $\Omega_G^{2r}(w_1', w_2')$. We require that $\Omega_G^{2r}(w_1', w_2')$ is at most $1/k$ times of average RTT of all normal links. In normal links, let RTT between each node pair (i, j) be p. Let the RTT between the wormhole peer be $p/2$ as wormhole nodes hide the tunneling delay. Therefore, we can say $p > p/2$. Let, for each node i, the average RTT with all its neighbours whose $d \leq 2r$ be denoted as $\chi_G^{N(i)}$, where $N(i)$ is the neighbours of node i in network G. Substituting $p/2$ with $\Omega_G^{2r}(w_1', w_2')$ and setting $k = 2$, we check the condition $\Omega_G^{2r}(w_1', w_2') \leq \chi_G^{N(i)}/2$ for the detection of short path wormhole links. When $k = 2$, the attacker connected by short path is very likely to be detected.

This parameter is used to detect short path wormhole tunnel. Let the RTT between the normal node pair (i, j) be p. Since the wormhole nodes hide the tunneling delay while propagating the packet, we may assume that the time to tunnel the packet between wormhole nodes connected by short path is less than p. Hence, RTT between the wormhole pair is considered to be p/k, where $k = 2$. Setting $k > 2$ will reduce the tunneling delay to such an extent that it may not be practical in a real short path wormhole tunnel scenario.

5.1. Time Complexity Analysis. Range-based ordinal MDS method for detecting wormholes has several steps: Floyd-Warshall shortest path algorithm, ordinal MDS method, and wormhole detection using complete bipartite subgraph. The time complexity has been analyzed as follows.

(i) The time complexity of Floyd-Warshall shortest path algorithm is $\mathcal{O}(v^2)$.

(ii) The time complexity of ordinal MDS is $\mathcal{O}(v^3)$.

(iii) Algorithm 3 comprises two parts. In the first part, we consider finding connected components in the set of suspect nodes. Let us consider the suspect nodes to be $w_0, w_1, w_2, \ldots, w_i \in S$. The time complexity for finding a connected component in the suspect node set S, that is, a path/edge from node w_1 to w_2, is $\mathcal{O}(we)$ since each path or edge is obtained in $\mathcal{O}(e)$ time [32]. Thus, the time complexity for finding each complete bipartite subgraph is $\mathcal{O}(we)$.

In the second part, Algorithm 3 needs to find a common neighbour of (w_1, w_2) for each connected node pair (w_1, w_2). Since the search is restricted to only one common neighbour for each wormhole pair, the number of common neighbour is a constant c. Thus, the time complexity of finding common neighbour is $\mathcal{O}(cn)$, that is, $\mathcal{O}(n)$.

6. Experimental Analysis

To verify the efficacy and performance of our method, experiments are conducted under different node distributions, radio models, and positions of wormholes.

6.1. Simulation Environment

6.1.1. Node Deployment. We have chosen two node deployment models: random deployment and perturbed grid deployment. In perturb grid model, the nodes are deployed on a grid $a \times b$. Each node in the network is perturbed with a perturb ratio p with the node's initial position. Each cell is a square grid with edge length l. Then, nodes having coordinate (x, y) are perturbed with $p = 0.2$ and are placed in the region

$[x-pl, x+pl]\times[y-pl, y+pl]$. In random deployment, nodes are placed randomly in the network field. It has been observed that in perturbed grid deployment the nodes' positions show more uniformity than the random deployment.

6.1.2. Communication Model. In our method, we have adopted UDG as the connectivity model. In UDG model, each pair of nodes i and j has bidirectional link if and only if their distance is no larger than r, where r is the communication radius.

6.1.3. Wormhole Placement. The first pair of wormhole node is placed randomly in the network. The other pairs are placed at uniform distances from the first pair. We have observed that if all wormhole nodes are placed randomly, they do not attract much traffic due to isolation. We have considered varying number of wormhole nodes with varying node density.

6.2. Simulation Results. We conduct experiments under various node densities and compare them with TRM [1], WORMEROS [4], and MDS-based local connectivity [7] methods and present the results. We use ns-2 simulator to implement our algorithm for performance evaluation. We deploy 100 nodes over a square field considering three network dimensions 10×10, 500×500, and 1000×1000. The initial transmission range of each node is set to 250 m. But, later, it is expanded to 500 m. The average node degree varies from 4 to 20. The maximum number of wormhole nodes is 20. For long path wormholes, our detection algorithm considers the fact that the colluding wormholes are not less than 6 hops apart from each other.

6.3. Detection Rate. Figures 3(a), 3(b), and 3(c) depict the relationship between varying number of wormholes and detection performance under three network distribution fields considering the UDG connectivity model and random deployment.

The results in Figures 3(a), 3(b), and 3(c) show that our method has almost 100% detection rate under different network field size with varying wormhole nodes and varying node density. In Figure 3(a), TRM method has slightly better detection rate when the network is dense. But Figures 3(b) and 3(c) show a sharp decline in detection performance of around 71% and 63%, respectively, when the network changes from dense to sparse. As TRM method solely relies on neighbour mismatch, the detection rate fails when the node density decreases. On the other hand, our approach uses the complete bipartite subgraph algorithm to clearly identify all the adversaries and, hence, the method of detection is suitable for both sparse and dense networks.

Figure 3(d) shows that perturbed grid deployment introduces some isolated wormholes which remain undetected and hence leaves an impact on the wormhole detection rate. It can be observed, from Figure 3(d), that the percentage of isolated wormholes gradually reduces to zero when the number of wormholes increases. Moreover, reduction in the number of isolated wormholes improves the detection efficiency because increasing number of wormholes establish pairwise wormhole links which are identified easily using the complete bipartite subgraph methods.

6.4. False Positive and False Negative. We conduct this experiment to examine the false positive and false negative rate with varying network density ranging from 20 to 100 nodes distributed randomly and in perturbed grid in a constant field size of 1000×1000. Figures 4(a) and 4(b) show the false positive rate in relation to the average node degree for random distribution and perturbed grid distribution, respectively. Figures 4(a) and 4(b) indicate that the number of false positives decreases as the node degree increases. Figure 4(a) shows that our method has greatly surpassed WORMEROS method and MDS-based local connectivity method in random deployment. In Figure 4(b), for perturbed grid distribution, the number of false positives in our method is less than MDS-based local connectivity method. However, we observe that initially when there are a less number of nodes, say 20 to 40, with average node degree from 4 to 8, our method shows few false positives. The reason is that some true nodes are suspected as probable wormholes while detecting short path wormhole links. As the node size increases, there is almost no false positives.

The result obtained in Figure 5 shows that in our approach the false negative is reduced to zero thus achieving detection rate of almost 100% in both sparse and dense networks while WORMEROS method has a detection rate of 92% only in dense networks with few false negatives.

6.5. Cost Analysis. The proposed long path wormhole link detection method comprises three algorithms, namely, Algorithms 1–3. We have calculated the number of iterations that take place in these three algorithms and computed their total. This total number is considered to be the cost, c, of our detection method. We have considered random deployment. We have executed our detection method a number of times by varying number of nodes and number of wormholes. For a fixed number of node and a fixed number of wormholes, the detection method is run a number of times over random deployments. The cost plotted in Figure 6 is the average of cost of all such runs. Figure 6 shows the cost of detecting wormholes with respect to the number of nodes and the number of wormholes in the network. When the number of nodes increases, detection cost becomes higher. If we keep the number of nodes fixed and vary the number of wormholes, cost increases but the rate of increase is less. When the number of wormhole nodes is 4, then the curve for cost, c, versus number nodes, n, may be approximated by a function $c = \mathcal{O}(n^{1.6})$. Thus, the solution is scalable.

6.6. Observation on Node Degree. Figure 7 shows that the presence of wormholes increases the average node degree by creating false neighbours. It is observed that wormhole-free environment shows a lower average node degree ranging from 4 to 15 while perturbed grid shows a relatively medium average node degree ranging from 10 to 25. However, random deployment shows high range of average node degree of 15 to 27. Thus, it could be inferred that wormhole attack creates an

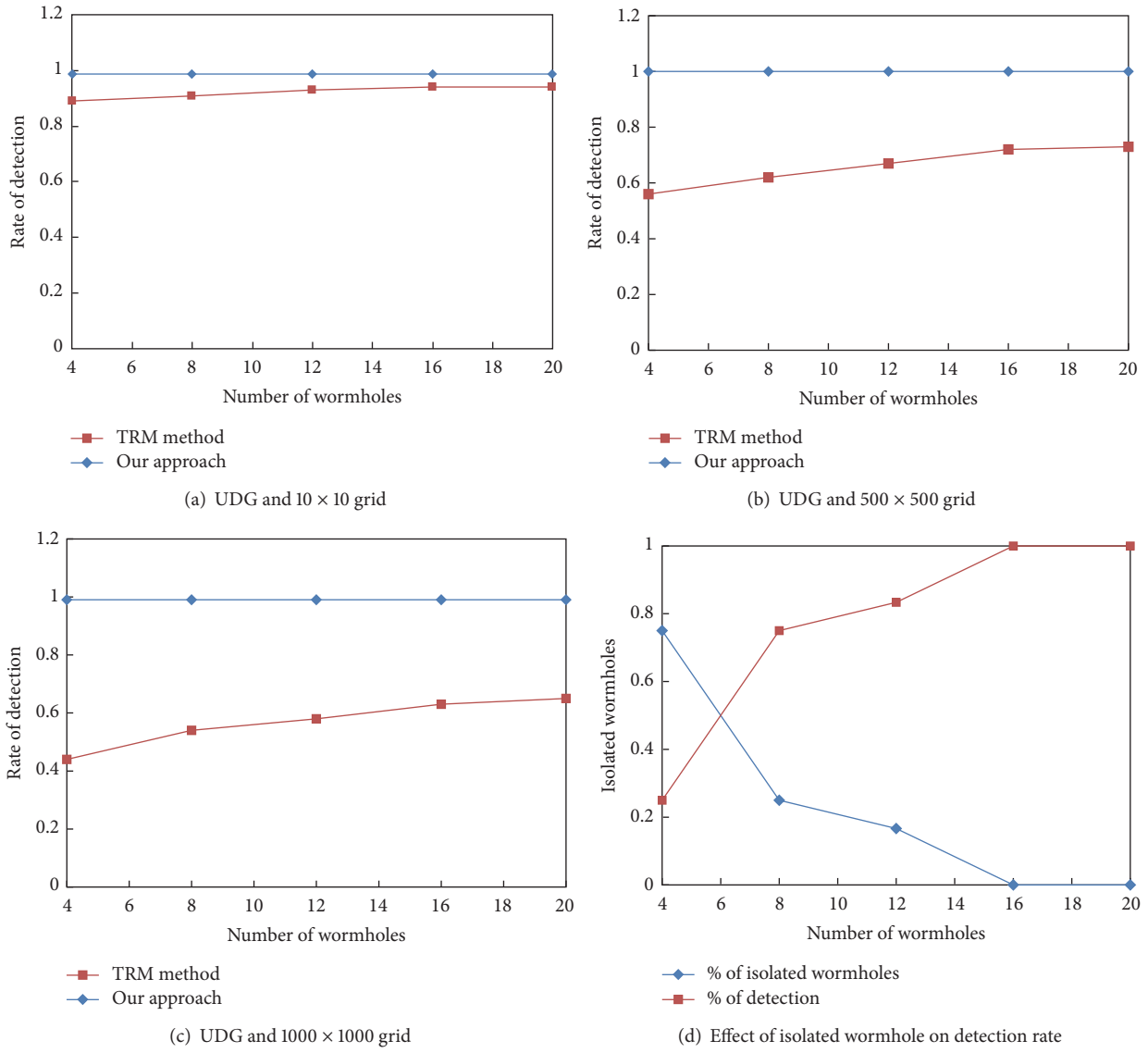


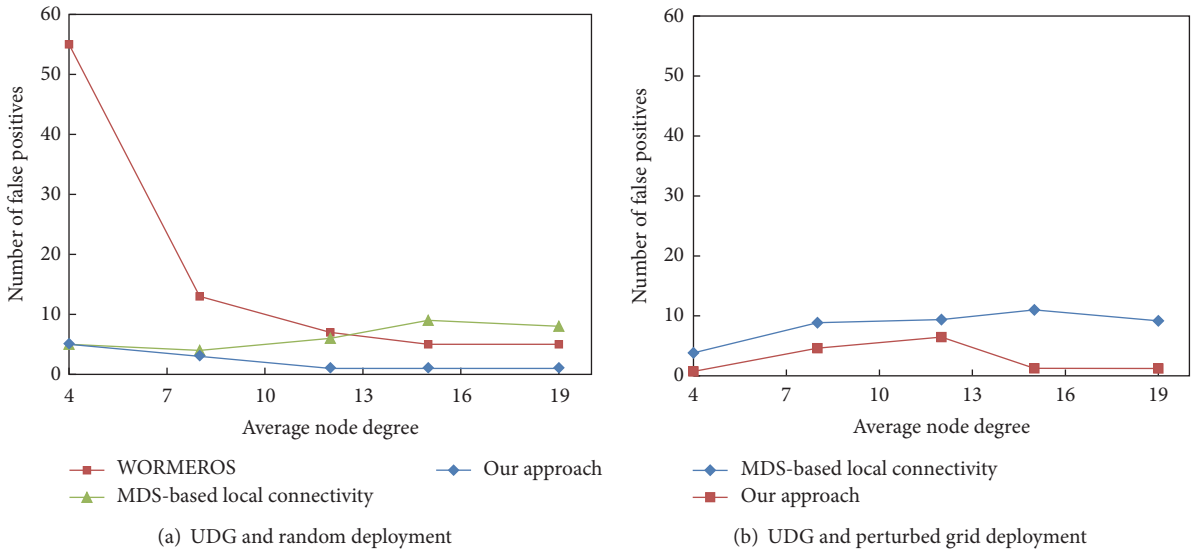
FIGURE 3: (a)–(c) show detection efficiency of 20 wormhole nodes in random deployment with varying network field size. (d) Detection efficiency of 20 wormhole nodes in perturbed grid deployment in 1000 × 1000 grid.

illusion of false neighbours in the network in order to attract large volume of traffic.

6.7. Multiple Wormholes. Figure 8 presents three scenarios of multiple wormholes placed at different position of the network. Figures 8(a) and 8(b) show the wormhole links which are far away from each other. Our method can detect all wormhole nodes connected by long path with almost no false positives. Figure 8(c) depicts a scenario where both ends of a wormhole pair are connected by a long path. One end of such wormhole pair is adjacent to one of the ends of another wormhole pair. These adjacent wormhole nodes eventually establish a short path wormhole link between each other. In our approach, such wormholes which are closely positioned to each other can be detected but with some false positives. The reason is that there might be some normal node pair (i, j) which is closely positioned. The RTT of such node pair is much less than the average RTT value of the network due to its proximity. The RTT value between such normal node pair is very close to abnormally low RTT value and, thus, this node pair (i, j) gets enlisted as wormholes too. As the sensor nodes generally maintain a distance from each other, such closely placed nodes are rare.

7. Conclusion

For the past few years, wormhole attacks have drawn more attention since they partition the network in two sets and disrupt the normal network functionalities. However, in earlier works, many countermeasures are proposed but those methods may require special hardware and/or suffer from high overhead. In this work, we analyze the topological differences



(a) UDG and random deployment (b) UDG and perturbed grid deployment

Figure 4: False positive rate with average node degree in (a) and (b).

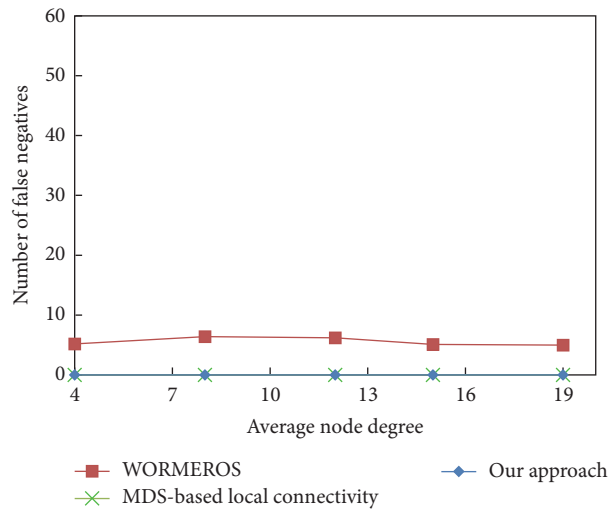


Figure 5: False negative rate with average node degree in UDG model with random deployment.

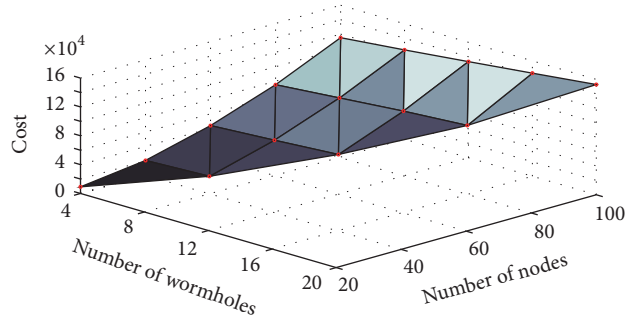


Figure 6: Detection cost in random deployment.

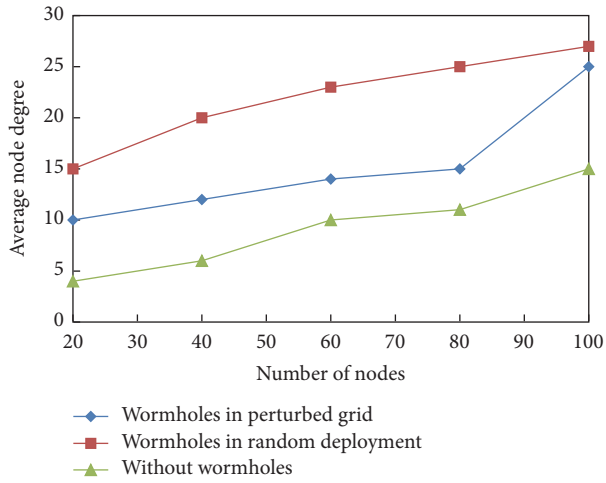


Figure 7: Effect of wormholes on average node degree.

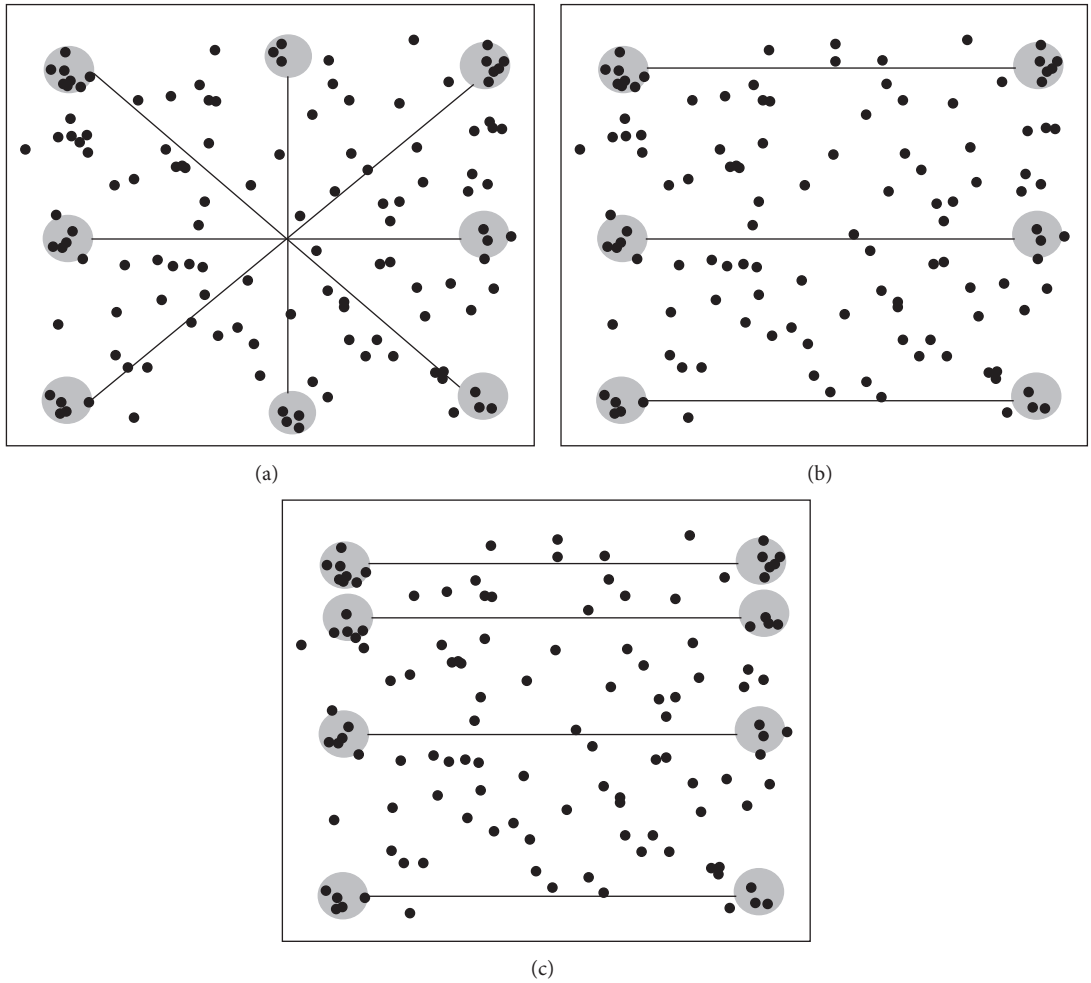


Figure 8: Multiple wormholes are placed at different positions of the network. (a) and (b) show wormhole links connected by long paths. Wormhole links connected by short path are shown in (c).

induced by wormholes and propose ordinal MDS-based network reconstruction using RTT to detect wormhole links. Our method can detect multiple wormhole links connected by short path and long path. The simulation results demonstrate that our approach can detect all wormhole nodes in dense as well as in sparse networks with perturbed grid and random deployment. Detection of long path wormhole links with almost no false positives has been achieved. Detection of wormholes connected by short paths introduces some false positives. So, in future, the issue of false positive for short path wormhole links remains open for further investigations.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

References

[1] G. Wu, X. Chen, L. Yao, Y. Lee, and K. Yim, "An efficient wormhole attack detection method in wireless sensor networks," *Computer Science and Information Systems*, vol. 11, no. 3, pp. 1127–1141, 2014.

[2] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 3, pp. 1976–1986, April 2003.

[3] L. Hu and D. Evans, "Using directional antennas to prevent wormhole attacks," in *Proceedings of the Network and Distributed System Security Symposium*, pp. 1–11, San Diego, Calif, USA, February 2004.

[4] H. Vu, A. Kulkarni, K. Sarac, and N. Mittal, "Wormeros: a new framework for defending against wormhole attacks on wireless ad hoc networks," in *Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications*, pp. 491–502, Springer, 2008.

[5] M. R. Alam and K. S. Chan, "RTT-TC: a topological comparison based method to detect wormhole attacks in MANET," in *Proceedings of the 12th IEEE International Conference on Communication Technology (ICCT '10)*, pp. 991–994, IEEE, Nanjing, China, November 2010.

[6] X. Ban, R. Sarkar, and J. Gao, "Local connectivity tests to identify wormholes in wireless networks," in *Proceedings of the 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '11)*, ACM, Paris, France, May 2011.

[7] X. Lu, D. Dong, and X. Liao, "MDS-based wormhole detection using local topology in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 145702, 9 pages, 2012.

[8] R. Maheshwari, J. Gao, and S. R. Das, "Detecting wormhole attacks in wireless networks using connectivity information," in *Proceedings of the IEEE 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 107–115, May 2007.

[9] W. Wang and A. Lu, "Interactive wormhole detection in large scale wireless networks," in *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST '06)*, pp. 99–106, November 2006.

[10] S. Čapkun, L. Buttyán, and J.-P. Hubaux, "SECTOR: secure tracking of node encounters in multi-hop wireless networks," in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 21–32, ACM, Washington, DC, USA, October 2003.

[11] H. S. Chiu and K.-S. Lui, "DePHI: wormhole detection mechanism for ad hoc wireless networks," in *Proceedings of the 2006 1st International Symposium on Wireless Pervasive Computing*, pp. 1–6, IEEE, Spa Phuket, Thailand, January 2006.

[12] Z. Tun and A. H. Maw, "Wormhole attack detection in wireless sensor networks," *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 2, no. 10, pp. 2184–2189, 2008.

[13] I. Khalil, S. Bagchi, and N. B. Shroff, "LITE WORP: a lightweight countermeasure for the wormhole attack in multihop wireless networks," in *Proceedings of the International Conference on Dependable Systems and Networks (DSN '05)*, pp. 612–621, Yokohama, Japan, July 2005.

[14] I. Khalil, S. Bagchi, and N. B. Shroff, "MOBIWORP: mitigation of the wormhole attack in mobile multihop wireless networks," *Ad Hoc Networks*, vol. 6, no. 3, pp. 344–362, 2008.

[15] S. Choi, D.-Y. Kim, D.-H. Lee, and J.-I. Jung, "WAP: wormhole attack prevention algorithm in mobile ad hoc networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC '08)*, pp. 343–348, Taichung, Taiwan, June 2008.

[16] R. Poovendran and L. Lazos, "A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks," *Wireless Networks*, vol. 13, no. 1, pp. 27–59, 2007.

[17] Z. Zhao, B. Wei, X. Dong, L. Yao, and F. Gao, "Detecting wormhole attacks in wireless sensor networks with statistical analysis," in *Proceedings of the WASE International Conference on Information Engineering (ICIE '10)*, vol. 1, pp. 251–254, August 2010.

[18] L. Lu, M. J. Hussain, G. Luo, and Z. Han, "Pworm: passive and real-time wormhole detection scheme for WSNs," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 356382, 16 pages, 2015.

[19] W. Wang and B. Bhargava, "Visualization of wormholes in sensor networks," in *Proceedings of the 3rd ACM Workshop on Wireless Security*, pp. 51–60, ACM, October 2004.

[20] D. Dong, M. Li, Y. Liu, X.-Y. Li, and X. Liao, "Topological detection on wormholes in wireless ad hoc and sensor networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 6, pp. 1787–1796, 2011.

[21] H. Chen, W. Lou, Z. Wang, J. Wu, Z. Wang, and A. Xi, "Securing DV-Hop localization against wormhole attacks in wireless sensor networks," *Pervasive and Mobile Computing*, vol. 16, pp. 22–35, 2015.

[22] H. Chen, W. Chen, Z. Wang, Z. Wang, and Y. Li, "Mobile beacon based wormhole attackers detection and positioning in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 10, no. 3, Article ID 910242, 2014.

[23] M. C. Van Wezel and W. A. Kosters, "Nonmetric multidimensional scaling: neural networks versus traditional techniques," *Intelligent Data Analysis*, vol. 8, no. 6, pp. 601–613, 2004.

[24] V. D. M. Nhat, D. Vo, S. Challa, and S. Lee, "Nonmetric MDS for sensor localization," in *Proceedings of the 3rd International Symposium on Wireless Pervasive Computing (ISWPC '08)*, pp. 396–400, IEEE, Santorini, Greece, May 2008.

[25] C. Miao, G. Dai, K. Mao, Y. Li, and Q. Chen, "RIMDS: multidimensional scaling iterative localization algorithm using RSSI in wireless sensor networks," in *Proceedings of the China*

Conference on Wireless Sensor Networks, pp. 164–175, Springer, 2014.

[26] B. Li, W. Cui, and B. Wang, "A robust wireless sensor network localization algorithm in mixed LOS/NLOS scenario," *Sensors*, vol. 15, no. 9, pp. 23536–23553, 2015.

[27] V. K. Chaurasiya, N. Jain, and G. C. Nandi, "A novel distance estimation approach for 3D localization in wireless sensor network using multi dimensional scaling," *Information Fusion*, vol. 15, no. 1, pp. 5–18, 2014.

[28] X. Zhang, Y. Wu, and X. Wei, "Localization algorithms in wireless sensor networks using nonmetric multidimensional scaling with RSSI for precision agriculture," in *Proceedings of the 2nd IEEE International Conference on Computer and Automation Engineering (ICCAE '10)*, vol. 5, pp. 556–559, Singapore, February 2010.

[29] F. Groenen, J. Patrick, and M. Velden, *Multidimensional Scaling*, Wiley Online Library, 2005.

[30] W. Härdle and L. Simar, *Applied Multivariate Statistical Analysis*, vol. 22007, Springer, Berlin, Germany, 2007.

[31] D. Eppstein, "Arboricity and bipartite subgraph listing algorithms," *Information Processing Letters*, vol. 51, no. 4, pp. 207–211, 1994.

[32] Y. Zhang, C. A. Phillips, G. L. Rogers, E. J. Baker, E. J. Chesler, and M. A. Langston, "On finding bicliques in bipartite graphs: a novel algorithm and its application to the integration of diverse biological data types," *BMC Bioinformatics*, vol. 15, no. 1, article 110, 2014.