IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 26, 2018, accepted January 5, 2019, date of publication January 24, 2019, date of current version February 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2894637

CREDND: A Novel Secure Neighbor Discovery Algorithm for Wormhole Attack

XIAO LUO[1], YANRU CHEN[2], MIAO LI[2], QIAN LUO[1], KANG XUE[1], SHIJIA LIU[2], AND LIANGYIN CHEN[2,3], (Member, IEEE)

[1]Second Research Institute of General Administration of Civil Aviation of China, Chengdu 610042, China
[2]College of Computer Science, Sichuan University, Chengdu 610065, China
[3]Institute for Industrial Internet Research, Sichuan University, Chengdu 610065, China

Corresponding authors: Shijia Liu (shijialiu@stu.scu.edu.cn) and Liangyin Chen (chenliangyin@scu.edu.cn)

Xiao Luo, Yanru Chen, and Miao Li contributed equally to this work.

This work was supported in part by the NNSFC&CAAC under Grant U1533203, in part by the Collaborative Innovation of Industrial Cluster Project of Chengdu under Grant 2016-XT00-00015-GX, in part by the National Natural Science Foundation of China under Grant 61373091, in part by the Science and Technology Department of Sichuan Province under Grant 19ZDYF0045 and Grant 19CXTD0005, and in part by the National key R&D Program of China under Grant 2018YFB1601200 and Grant 2018YFB1601201.

ABSTRACT As nodes' characteristics that they are self-governed and resource-limited, wireless sensor networks (WSNs) face potential threats due to various attacks, among which the most threatening attack is wormhole attack. Wormhole attack severely imperils WSNs and is difficult to be detected, for it causes incorrect routing by private tunnels and damages to WSNs in terms of data leakage, data dropping, and delayed delivery. However, the existing solutions are based on additional hardware, incur high communication overhead, or fail to give consideration to all types of wormholes. In this paper, we propose CREDND, a protocol for creating a Credible Neighbor Discovery against wormholes in WSN, which can detect not only external wormholes through the hop difference between the own exclusive neighbors but also internal wormholes through enabling the common neighbor nodes as witnesses to monitor whether the authentication packets are forwarded by malicious nodes. CREDND is a simple, localized protocol and needs no special hardware, localization, or synchronization, but it improves the ability of wormhole defense. The simulation results are provided, showing that CREDND outperforms in wormhole detection than other same types of solutions.

INDEX TERMS Secure neighborhood, neighbor discovery, network security, wireless sensor networks, wormhole attack.

I. INTRODUCTION

Recently, a great variety of pervasive technologies, e.g. intelligent sensing [1], low-power processing [2], [3], and wireless communication [4], [5], offer unprecedented opportunities to develop wireless sensor networks (WSN). With the advantages of low cost, large scale, densely distributed deployment and self-configuration [6], WSNs are widely used for industrial control, logistics management, environmental monitoring, military defense, and civilian life [7], [8], most of which support range of time and data sensitive applications and have high requirements for safety. However, as a consequence of distinctive open nature, limited resource availability, infrastructure less and self-governing nature, WSNs are very delicate to many attacks [9], [10].

Wormhole attacks are among the most severe and sophisticated security threats to WSNs routing protocols [11], [12], where malicious nodes are placed strategically to distort the network topology and tunnel packets selectively using the false established routes. Wormhole detection and prevention are very challenging issues [13], [14]. The wormhole attacks can be executed by the external nodes (who only forward packets and do not process the cryptographic) or internal nodes (the compromised nodes inside the network who process packets just like other normal nodes) [15], of which the internal attackers are more dangerous and difficult to detect [16]. However, Chen *et al.* [6] hold the view that the wormhole attack is a typical external attack. Moreover, the great majority works of literature usually pay excessive attention to the external wormholes but ignore internal wormholes that are also common in WSNs.

For example, as shown in Fig. 1, the nodes A and B are not neighbor of each other, thus the normal path will be $A-E-F-G-H-B$ with a hop count of 4. But in Fig. 1(a), nodes A and B seem to be direct neighbor under the influence that the external wormhole nodes M_1 and M_2 between two distant locations tunnel the packets through an out-of-band channel.

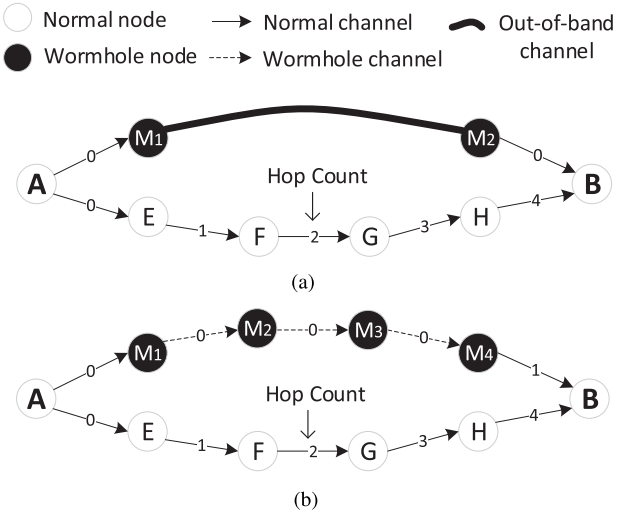

FIGURE 1. The different attacks launched by (a) External wormhole and (b) Internal wormhole.

On the other side, internal wormhole nodes M_1, M_2, M_3, and M_4 establish a virtual tunnel in Fig. 1(b), the packets of node A will be encapsulated by node M_1 and tunneled to node B with a hop count of 1. No matter external wormhole or internal wormhole it is, wormhole tunnels give two distant nodes an illusion that they are very much closer than it actually is to each other [17], so node A will choose the shorter wormhole route rather than the correct route. After getting involved in the route, the wormhole attackers launch a variety of sabotages such as cryptanalysis, protocol identification, selective dropping, eavesdropping, and replay attack [18], which significantly jeopardize routing, localization and other functions in WSN [18]–[20]. Its characteristics and impact on networks are described in [21]–[23] cites.

Therefore, a technique that can find wormhole-free routes in the network is required. Some proposed solutions depended on cryptographic techniques [24], [25], but they consume higher energy for computationally complex cryptography [26], [27]. Some wormhole preventive approaches are based on geographical location which needs equipping with GPS [28], directional antennas [29] or mobile beacon [30], incur high cost and communication overhead in WSNs. There are also some location-based approaches [6], [31] do not need additional hardware, but their performance suffers badly when packet loss and communication range of the nodes are not identical. Other approaches are based on time analysis, need accurate time measurements or global tight clock synchronization [32]–[36] where the data availability is not ensured. Connectivity-based algorithms attract more and more attention for they detect wormholes without additional hardware. Some connectivity-based solutions based on ideal assumptions [16], [37], [38] which are difficult to implement. Some connectivity-based solutions are the statistical solutions [39]–[42] require extra processing and cause the delay in communication. And some are distributed approaches using neighborhood table [43]–[46] whose wormhole defensive ability is not comprehensive enough.

In view of the shortcomings of current approaches, we propose a novel Credible Neighbor Discovery algorithm named CREDND for wormhole detection, which can not only detect wormholes without additional hardware but also detect wormholes whatever type the wormhole is. CREDND achieves better performance and less energy consumption. The main contributions of this paper are summarized as follows.

- We propose CREDND, a novel secure neighbor discovery algorithm for wormhole detection, using hop difference and local monitoring. CREDND is able to detect both external wormholes and internal wormholes. It improves the ability of wormhole defense and saves node energy at the same time.
- We also propose the concept of *Neighbor Ratio Threshold*, which avoids performing wormhole detections for all nodes in WSN, contributing to improving the accuracy of wormhole detection and saving energy.
- We test the accuracy of CREDND by simulating and comparing it with two other wormhole detection algorithms, SECUND [45] and SEDINE [46], who also use hop difference and local monitoring. The results demonstrate that our approach outperforms existing approaches.

The rest of the paper is organized as follows. We first review related work in Section 2 and then introduce background knowledge in Section 3. In Section 4, we introduce the detailed design of our CREDND. Then section 5 presents the evaluation results of system performance. Finally, we conclude our work and discuss the future work in Section 6.

II. RELATED WORK

This section summarizes related work and generally divides previous researches on wormhole attacks into four categories according to the different resources they use: cryptographic-based approach, location-based approach, time-based approach, and connectivity-based approach.

A. CRYPTOGRAPHIC-BASED APPROACH

Some proposed solutions depended on cryptographic techniques to protect routing packets and detect wormholes. Sookhak *et al.* [24] presented an approach where malicious nodes were detected based on identifying the best reliable neighbors using the pairwise key pre-distribution technique. Vo *et al.* [25] described a novel multi-level authentication model and protocol (MLAMAN) which allows all intermediate nodes to authenticate control packets on a hop-by-hop basis and at three levels. It is quite well known that cryptographic-based schemes consume higher energy in order to process the computationally complex cryptography for securing routing protocol [26], [27].

B. LOCATION-BASED APPROACH

Wang and Wong [28] proposed an end to end wormhole detection method. After knowing each other's position, the source node detects wormholes by comparing the actual

hops and the estimated hops. Although it can secure the neighborhood effectively under the condition that the location relationships of nodes can be safely notified to other nodes, the fact that each node is required to be equipped with GPS or employ some other positioning technology, such as directional antennas as in [29] and mobile beacon as in [30]. It not only increases the hardware cost but also impacts the node's energy consumption. Chen *et al.* [6] described the DV-Hop localization mechanism that uses the label to provide secure location accuracy. The beacon nodes are distinguished and labeled according to their geographic relationships, then DV-Hop localization is applied to the labeled neighboring nodes, after avoiding the wormhole links. Wang *et al.* [31] also proposed a modified distance vector based DV-Hop localization algorithm to detect attacks using the average hop distance between beacon nodes and unknown nodes. These location-based approaches do not need additional hardware, but their performance suffers badly where packet loss and communication range of the nodes are not identical.

C. TIME-BASED APPROACH

Time-based approaches are based on the time interval between the packets sending and receiving. For example, Kaur *et al.* [32] proposed a technique which identifies the wormhole links by calculating the maximum end to end delay between two nodes within the communication range. Mukherjee *et al.* [33] reconstructed neighborhood based on computed round trip time (RTT) between node pairs and used topology change information to detect wormholes. Amish and Vaghela [34] proposed a method uses the RTT of every route and threshold value to detect the wormhole link. Karlsson *et al.* [35] presented a wormhole detection algorithm which identifies time measurement tampering in traversal time and hop-count analysis. Shi *et al.* [36] described a time-based approach in which after the route discovery process, the source node estimates the hop count on the basis of time. These schemes are based on accurate time measurements or require the nodes to have tightly synchronized clocks, which is difficult to implement and resource-consuming. In addition, the data availability is not ensured, because it is impossible to detect the wormholes through which the transmission time of the packets is just within the allowed time delay, for the reason that the MAC protocol may also cause some unpredictable delays. More importantly, a packet suffers only the propagation delay which could be small for wormholes using high-speed links.

D. CONNECTIVITY-BASED APPROACH

Connectivity-based algorithms can detect wormholes without additional hardware, which is the reason why it attracts more and more attention. Wu *et al.* [37] proposed an algorithm that detects wormholes by checking whether the one-hop neighbors of all nodes come to be its direct neighbors after doubling the communication range of each node through beacon messages, but the fact is that the doubled communication range of all nodes cannot be guaranteed due to the environmental impact, and the communication overhead is increased since every pair has to do this periodically. Ho and Wright [38] presented a solution using sequential analysis with software attestation to detect malicious nodes. Here the adversary needs to compromise some nodes initially to infect them with self-propagating malicious codes which may propagate through the network to cause damage. Khan *et al.* [16] showed a distributed detection scheme where special nodes called DPS (Detection and Prevention System) nodes were statically employed to detect malicious nodes. It needs incorporation of special guard nodes which will undoubtedly increase the cost of WSN. Above solutions are based on ideal assumptions and difficult to implement.

Some solutions are the statistical solutions, which try to detect wormholes by analyzing different factors that can be used. Lu *et al.* [39] proposed the method that first attempt toward creating a graph theoretical approach, called Worm Planar, who just utilizes localized connectivity information and is capable of capturing the global require symptoms of wormholes directly in the wireless networks. Akilarasu and Shalinie [40] showed an approach for wormhole detection based on Finite State Machine (FSM) and priority mechanism. Jamali and Fotohi [41] proposed an improvement over AODV routing protocol called defending against wormhole attack (DAWA) employs the fuzzy logic system and artificial immune system to defend against wormhole attacks. Tiruvakadu and Pallapa [42] proposed a Wormhole Attack Confirmation (WAC) System using honeypot. It built a Wormhole Attack Tree based on its symptoms. The honeypot confirms the wormhole attack using the Wormhole Attack Tree (WAT) and history of attacks. Such solutions require extra processing and cause a delay in communication.

There are also a number of distributed approaches using neighbor information to detect abnormalities as discussed below. Ji *et al.* [43] proposed a distributed detection algorithm named DAWN, exploring the change in the flow directions of the innovative packets affected by wormhole nodes using local information that can be obtained from regular network coding protocols with reduced overheads. DAWN solely depends on Expected Transmission Count (ETX) to determine the attack, which can be manipulated by the attackers when the wormhole tunnel is built using the high-speed wired link. Giannetsos and Dimitriou [44] presented a decentralized neighbor-based method wherein the neighbors were attested as true neighbors after a k-hop connectivity test. But a wormhole attacker can still disrupt the network if the attacker collects the hello packets at one end of the tunnel and broadcasts it at the other end, and discovering and maintaining k-hop neighbors for each node is an overhead. Hayajneh *et al.* [45] proposed SECUND to detect wormholes by using hop differences, which detects wormholes by checking whether the hop distance between the exclusive neighbors exceeds the predefined threshold. These solutions performed badly in the wormholes with short hops. Hariharan *et al.* [46] described SEDINE which detects wormholes relies on the overhearing capability of nodes to detect whether a packet

is being forwarded. But its defensive ability is not comprehensive enough too, as they can't defend against wormholes who tunnel packets using the out-of-band channel.

III. PRELIMINARIES

In this section, we first introduce wormhole attacks. Then we specify the node distribution characteristics in WSNs.

A. WORMHOLE ATTACKS

In WSNs, the most severe threat is wormhole attacks [10] [11]. Generally, this type of attack is launched by two or more malicious nodes having a private channel, called a tunnel, between them. Four types of tunnel, packet encapsulation, out-of-band, high power transmission, and packet relay are summarized in [12] and [42]. According to the different malicious nodes participating in these tunnels, we divide wormhole attacks into external wormholes and internal wormholes. External wormholes can distort network behaviors without obtaining the system's authorization, while internal wormholes are authenticated and thus more devastating to the security of the system [6].

1) EXTERNAL WORMHOLE

External wormhole attackers use the different medium from normal nodes, e.g. out-of-band channel and high transmission mode, so external wormhole is initiated by malicious nodes in Hidden Mode (HM). In HM, malicious nodes are hidden from normal nodes, which on receiving a packet they simply forward the packet without processing it, as the malicious nodes in HM do not have the communication key of the WSN. By doing so, the malicious nodes are invisible and external for WSN, for they never appear in the routing tables of normal nodes. Its only way of attack is getting right-of-way to attract a huge amount of traffic, as a result, routes from the sources to destinations that avoid external wormhole links are usually much longer than the routes that make use of external wormholes.

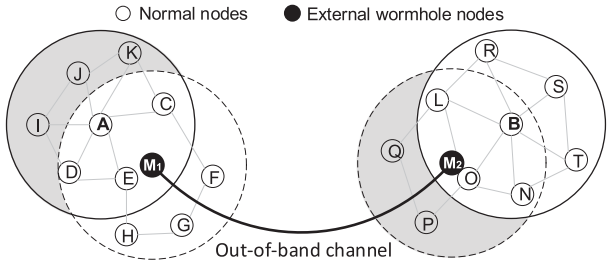

FIGURE 2. The wormhole formed by external malicious nodes.

Fig. 2 shows an external wormhole formed by two external malicious nodes M_1 and M_2, which connect with each other through an out-of-band channel. Node M_1 can tunnel any packets it received (such as the packets from node A, C, D, E, F, G, H) to node M_2, and in turn, node M_2 broadcasts the same packets, so that every node within M_2's coverage area (such as node B, L, N, O, P, Q) can receive them. Therefore, every node in the coverage areas of M_1 and M_2 (such as node A and node B) is under the illusion that they are neighbor, although they are separated by a large area geographically.

2) INTERNAL WORMHOLE

Internal wormhole attackers use the same medium as normal nodes, e.g. packet encapsulation and packet relay. The originators of internal wormholes are the nodes in the WSN (usually hijacked by the enemy) or the nodes that have the network key, so they operate in Participation Mode (PM). Some researchers believe that no normal node can be hijacked for the time of neighbor discovery (ND) is too short (usually a few seconds) [46], thus internal wormhole attack was generally neglected. In fact, this assumption does not hold when nodes are deployed incrementally, so we relax this assumption that only a few normal nodes are hijacked. In PM, malicious nodes are visible and internal for WSNs because they process and control packets just like other normal nodes. These malicious nodes appear in the routing tables of normal nodes and the hop count increase when packets are routed. Moreover, packets will be encapsulated, hence avoiding the increase in hop count between the wormhole link. On the one hand, there are only a few normal nodes be hijacked, on the other hand, the transmission time of the packet from the source to the destination may be time-out if the wormhole link length is excessively long. In a word, internal wormhole links that attack in PM are generally short.

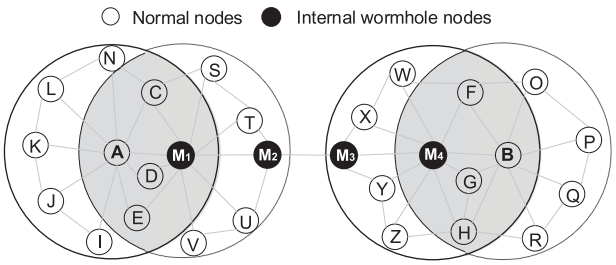

FIGURE 3. The wormhole formed by internal malicious nodes.

An internal wormhole formed by four internal malicious nodes is shown in Fig.3, and we will introduce the effect on the two ends of this internal wormhole. Internal wormhole nodes M_1, M_2, M_3, and M_4 establish a virtual tunnel, where node M_1 will encapsulate the packets it received (such as the packets from node A, C, D, E, U, V, S, T) as the new data packets and tunnel them to node M_4 with node M_2, M_3 only assumed as the relays of M_1, so the hop count will not increase at nodes M_1, M_2, M_3, and M_4. Node M_4 will rebroadcast the packets after extracting them from the data packets later. The above operations make it seems that there is only one node between node M_1, M_2, M_3, and M_4, and every node within node M_4's coverage area (such as node B, F, G, H, W, X, Y, Z) can receive the packets from node M_1's neighbors with a hop count of 1. Therefore, every node in the coverage areas of M_1 and M_4 (such as node A and node B) is under the illusion that they are one-hop-neighbor.

B. NODE DISTRIBUTION

There exist two types of deployment strategies namely deterministic and non-deterministic deployment [10]. Sensor nodes are placed manually and the data transmission takes place through the precompiled routes in the case of deterministic deployment approach. Whereas sensor nodes are scattered randomly in non-deterministic deployment technique without any precomputed paths. We consider wormhole detection under non-deterministic deployment, which is more challenging. Sensor nodes are randomly deployed in practical applications for lacking the prior knowledge of the target area. And it is generally believed that the nodes randomly deployed approximately conform to the Poisson distribution [47], which can be modeled as:

$$P(N(S)=k)=\frac{e^{-\lambda\|S\|}\left(\lambda\|S\|\right)^{k}}{k!},\quad k=0,1,\cdots \tag{1}$$

The above is the formula of Poisson distribution, where P denotes the probability, N denotes a functional relationship, S denotes an area in the WSN, k denotes the number of sensor nodes in S, $\|S\|$ denotes the area of S, e is the bottom of natural logarithm, and λ is the average density of sensor node distribution, that is, the number of sensor nodes in a unit area.

IV. ALGORITHM DESIGN

In this section, we describe how CREDND works. We will first specify some assumptions of our sensor network for better introduce CREDND. Then, we propose the concept of *Neighbor Ratio Threshold* to determine which node pairs need to be detected (Algorithm 1). Finally, we describe the detections of external wormholes (Algorithm 2) and internal wormholes (Algorithm 3) on these suspected node pairs, respectively.

A. ASSUMPTIONS

In order to better introduce our solution, we assume that all links are bi-directional, i.e. if node A hears node B, then node B also hears node A. It also assumes that all the nodes in the WSN have a similar communication range and omni-directional antennas. In our model, we allow packet losses to occur due to link errors or collisions. We default that CREDND is applied to a static WSN and there is no denial of service attacks that prevent two neighboring nodes from becoming neighbor, physical layer jamming attacks, and physical destruction of nodes.

B. NEIGHBOR RATIO THRESHOLD

It is a waste of energy to check all nodes whether be affected by wormholes, for fewer nodes are affected by wormholes than the entire network. The wormholes will increase the connectivity of the network and an obvious increase will be seen in the neighbor number. Therefore, the neighbor number of a node within the impact scope of the wormhole will be more than those not within. We propose the *Neighbor Ratio Threshold* to compare the neighbor number of a node with all its neighbors to avoid launch wormhole detections for all nodes in the WSN. Specific methods are as follows: at first,

Algorithm 1 *Neighbor Ratio Threshold* Is Used to Determine Which Nodes Need to be Detected

Input: The entire network W with nodes O and their neighbor set N, and *Neighbor Ratio Threshold* r
Output: The node pairs who need to be detected.

```
for each node o_i in O and its neighbor set N_i in N do
    Let n_i = |N_i| (the neighbor number of o_i);
    for each node o_j ∈ N_i do
        n_j = |N_j| (the neighbor number of o_j);
        Initialize s = 0;
        s = s + n_j;
    Calculate the average neighbor number of o_i's neighbors n̄_i = s / n_i;
    Calculate o_i's neighbor ratio r_i = n_i / n̄_i;
    if r_i > r then
        Add o_i to suspected nodes set S;
for each nodes s_i ∈ S do
    for each nodes s_j ∈ S do
        if s_i and s_j are neighbors then
            Perform external wormhole detection (Algorithm 2) for neighbor node pair s_i and s_j;
        if s_i and s_j have common neighbors then
            Perform internal wormhole detection (Algorithm 3) for node pair s_i and s_j;
```

every node in the WSN will know its neighbors after ND. Then, the node calculates the ratio of its neighbor number and the average neighbor number of all its neighbors, named neighbor ratio. Finally, the neighbor ratio will be compared with *Neighbor Ratio Threshold* to determine whether wormhole detection is required. We will list all nodes whose neighbor ratio exceeds the *Neighbor Ratio Threshold* as suspected nodes. In these suspected nodes, external wormhole detections are performed on direct neighbor node pairs, internal wormhole detections are performed on the rest node pairs who have common neighbors. All of these are taken into account as described in Algorithm 1.

According to our experiment, the default value of *Neighbor Ratio Threshold* (whose impact will be discussed in Section V) is set to be 1.2.

C. WORMHOLE DETECTION

After comparing the neighbor ratio of nodes with *Neighbor Ratio Threshold*, we can determine whether or not the wormhole detections are needed and on which node pairs. If needed, the next stage can be divided into two parts: external wormhole detection and internal wormhole detection.

1) EXTERNAL WORMHOLE DETECTION

External wormhole detections should be performed on direct neighbor node pairs in the listed suspected nodes. The main

Algorithm 2 External Wormhole Detection

Input: N_A: the neighbor set of node A; N_B: the neighbor set of node B; w: *Wormhole Threshold*.

Output: The result of whether there exists an external wormhole between node pair A and B

Let $E_A = N_A - N_A \bigcap N_B - \{B\}$ (the exclusive neighbor set of node A);
Let $e_A = |E_A|$ (the exclusive neighbor number of node A);
if $e_A \geq 2$ **then**
    for *each node* o_i *in* E_A **do**
        for *each node* o_j *in* E_A **do**
            Calculate the new hop count h_j from o_i to o_j which bypasses N_B;
            if $h_j \geq w$ **then**
                There exists an external wormhole between A and B, and they remove each other from their neighbor tables;
else if $e_A < 2$, *and the neighbor ratios of A* r_A*>1.5, and* r_B*>1.5* **then**
    There exists an external wormhole between A and B, and they remove each other from their neighbor tables;
else
    There is no external wormhole between A and B;

Algorithm 3 Internal Wormhole Detection

Input: N_A: the neighbor set of node A; N_B: the neighbor set of node B.

Output: The result of whether there exists an internal wormhole between node pair A and B

Let all nodes in $N_A \cup N_B$ cannot send messages;
Let A and all nodes in $N_A \bigcap N_B$ go to monitoring mode;
Node A sends an authentication packet to node B;
for *each node* $o_i \in (N_A \bigcap N_B)$ **do**
    Initialize $tag_{A\to B} = 0$, $s = 0$;
    if o_i *receives the authentication packet sent by node A and the reply from node B to A later* **then**
        $tag_{A\to B} = 1$;
    else if o_i *hears that the authentication packet has been forwarded* **then**
        $tag_{A\to B} = -1$;
    $s = s + tag_{A\to B}$;
if A *receives the reply from B within time* τ **then**
    There is no internal wormhole between A and B;
else if A *monitors that the authentication packet has been forwarded* **then**
    There exists an internal wormhole between A and B, and they remove each other from their neighbor tables;
else if A *does not hear any information within time* τ **then**
    if $s \geq 1$ **then**
        There is no internal wormhole between A and B;
    else if $s < 1$ **then**
        There exists an internal wormhole between A and B, and they remove each other from their neighbor tables;

principle of external wormhole detection is to check and compare the hop differences between their own exclusive neighbors. The basic algorithm is given in Algorithm 2, and the description of all the steps with some discussion is presented next.

In Fig.2, as we discussed above, node A and B mistake that they are neighbor, and the nodes in the communication range of the external wormhole node M_2 are mistakenly considered to be node A's neighbors due to the external wormhole, so node A's neighbor set is $N_A = \{B, C, D, E, I, J, K, L, N, O, P, Q\}$. Similarly, node B's neighbor set is $N_B = \{A, C, D, E, F, G, H, L, N, O, R, S, T\}$. Then we have the common neighbor of neighbor node pair A and B is $N_A \bigcap N_B = \{C, D, E, L, N, O\}$. Node A's exclusive neighbor set is $N_A - N_A \bigcap N_B - \{B\} = \{I, J, K, P, Q\}$ (the shaded part on Fig.2). We can know that, for node A, the maximum hop-count between any two nodes in its exclusive neighbor set $\{I, J, K, P, Q\}$ is 1; however, the fact is that $\{I, J, K\}$ and $\{P, Q\}$ are far apart from each other and the real hop between them is much larger than 1.

By the above observation, we can select a node whose exclusive neighbor number is greater than 2 between a neighbor node pair, and specify new links between its exclusive neighbors should bypass the other node's neighbors, then detect whether the hop counts of these new links are greater than the *wormhole threshold* (the calculation of *wormhole threshold* is defined below) to determine whether there exist external wormholes. We choose node A from the neighbor node pair A and B as an example, and specify that the link between node A's exclusive neighbor set $\{I, J, K, P, Q\}$ (the shaded part on Fig.2) should bypass node B's neighbor set $\{A, C, D, E, F, G, H, L, N, O, R, S, T\}$(the non-shaded part on Fig.2). It is thought that there exist external wormholes between node A and node B once the hop counts of these new links exceeds *wormhole threshold*, such as the new link from node I to P. Then, node A and B remove each other from their neighbor tables and broadcast the deleted neighbor nodes to their neighbors. Otherwise, there is no wormhole.

Note that SECUND [45] also uses hop difference to detect wormholes, but differs from CREDND, it detects the hop difference from a node's exclusive neighbors to the other node's exclusive neighbors, and also required to bypass the other node's direct neighbors. For an instance shown in Fig.2, SECUND will select all nodes in node A's exclusive neighbor

set $\{I, J, K, P, Q\}$ and count the hops from them to all nodes in node B's exclusive neighbor set $\{R, S, T, H, G, F\}$ with bypassing node B's other direct neighbors. It is evident that SECUND requires more computation, so we can conclude that the energy consumption of CREDND is lower than SECUND when calculating the hop difference.

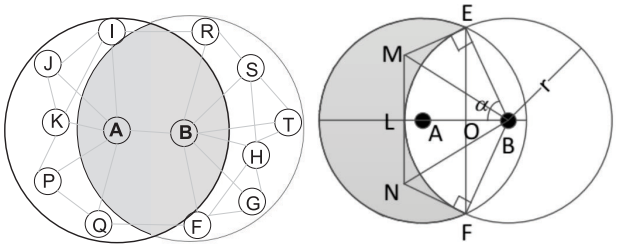

FIGURE 4. A normal neighbor relationship between node *A* and *B*.

a: CALCULATE WORMHOLE THRESHOLD

The *wormhole threshold* can be approximately calculated. Fig.4(a) shows a normal neighbor relationship between node A and B. As shown in Fig.4(b), of which the nodes in shadow part is node A's exclusive neighbor set, and the worst-case is that the two of node A's exclusive neighbors are in the vicinity of point E and point F, respectively. We can approximately calculate the hop count of the new link from node E to node F which cannot include B's direct neighbors by calculating the length of polyline from E through M, N to F.

As shown in Fig.4(b), node A and node B are direct neighbor, the distance between them is d $(0 < d \leq r)$, the midpoint of the line AB is O, and the length of the line BE equals node radius r, then we can easily calculate the length of the line EM is: $r \cdot \tan\left(\frac{1}{2}\arccos\frac{d}{2r}\right)$. By symmetry, the length of the polyline from E through M, N to F is: $4r \cdot \tan\left(\frac{1}{2}\arccos\frac{d}{2r}\right)$

So the hop count from node E to F is about:

$$f(d) = 4r \cdot \tan\left(\frac{1}{2}\arccos\frac{d}{2r}\right)/r = 4\tan\left(\frac{1}{2}\arccos\frac{d}{2r}\right) \tag{2}$$

The derivative of formula (2) is:

$$f'(d) = -\frac{1}{4r \cdot \left[\cos\left(\frac{1}{2}\arccos\frac{d}{2r}\right)\right]^2 \cdot \sqrt{1-\left(\frac{d}{2r}\right)^2}} \tag{3}$$

Formula (3) is a monotonically decreasing function. When d is infinitely close to 0, $f(d)$ is close to 4. When d is equal to r, $f(d)$ is minimum and roughly equals 3. So the default value of *wormhole threshold* (whose impact will be discussed in Section V) is set as 4.

b: SPECIAL CASES

In general, the method of using hop difference has been mentioned above can play its role well. However, there are two special cases, where external wormhole detections cannot be launched as there is no exclusive neighbor for neighbor node pairs.

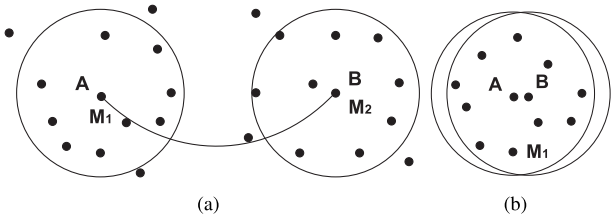

FIGURE 5. Special cases: (a) the wormhole node and normal node were deployed too close (b) two normal nodes were deployed too close.

As shown in Fig.5(a), external malicious node M_1 and normal node A are almost deployed at the same point, so are node M_2 and B, where there exists a wormhole. In this case, we cannot detect wormholes by hop difference for node A's exclusive neighbor set for its neighbor B is empty. But we can know that the neighbor numbers of node A and node B are approximately doubled compared to the nodes without wormhole affected by neighbor ratio, we still think there exists a wormhole and let node A and B remove each other from their neighbor tables.

On the other hand, As shown in Fig.5(b), node A and B were deployed too closed and an obvious increase of neighbor number will also be seen if there is an end of wormhole M_1 in their communication range. The judgment will be that there is a wormhole between node A and B and they remove each other from their neighbor tables, as *Neighbor Ratio Threshold*s are approximately doubled, although node A's and node B's exclusive neighbor sets are empty. But the fact is the opposite that node A and B are real neighbors and there is no wormhole between them. Such situation is usually relatively less and the detection result is equivalent to the case that node A and B do not succeed in finding each other, whose impact is relatively small compared to the case where the wormholes are really present but not detected.

2) INTERNAL WORMHOLE DETECTION

After the external wormhole detections, the external wormholes whose real links are larger than *wormhole threshold* will be successfully detected and removed. But internal wormholes and some external wormholes whose links are relatively short are still undetected. Therefore, this step is to deal with this situation.

According to the above description, internal wormhole detections should be performed on the rest node pairs who have common neighbors in the listed suspected nodes. The principle of internal wormhole detection is to enable suspected node pairs' common neighbors as witnesses to hear whether the packets between them are forwarded. The basic algorithm is given in Algorithm 3, and the description of all the steps with some discussion is presented next.

At first, the node pair ready for internal wormhole detection should obtain the verification right, where their neighbors cannot send messages during the verification so as to prevent the monitoring from being affected. Their common neighbors enter the monitoring mode and a node in the node pair send an authentication packet to the other

node. And the authentication packet will be sent again if the node that has sent the authentication packet hears irrelevant information.

As shown in Fig.3, node A and node B mistake that they are one-hop-neighbor due to internal wormhole nodes M_1, M_2, M_3, and M_4. Node A's neighbor set is mistaken as $N_A = \{B, C, D, E, F, G, H, I, J, K, L, N, W, X, Y, Z, M_1, M_3\}$ and node B's neighbor set is mistaken as $N_B = \{A, C, D, E, F, G, H, O, P, Q, R, S, T, U, V, M_2, M_4\}$. If the link between node pair A and B needs to be detected, the common neighbor set $N_A \bigcap N_B = \{C, D, E, F, G, H\}$ (the shaded part on Fig.3) can be determined and used to be witnesses for local monitoring. The neighbors of node A and B cannot send messages after the verification right is obtained. Node A and the nodes in $N_A \bigcap N_B$ (the shaded part of Fig.3) go to the monitoring state. Then node A sets the destination address of the authentication packet as the address of node B and sends it. Next, node A and witness nodes can hear one of three kinds of packets: the reply from node B to node A, the forwarded packet from node A, and the packets sent by other nodes. Also, some witness nodes may not receive any correlative packet, because they are not the real neighbors of node A and B.

For any witness node, they initialize their tags of link $A \rightarrow B$ as 0 at first. A witness node will set its tag of link $A \rightarrow B$ as 1 if it receives the authentication packet sent by node A and the reply from node B to A later. A witness node will set its tag of link $A \rightarrow B$ as -1 if it hears that the authentication packet has been forwarded; A witness node's tag of link $A \rightarrow B$ will still be 0 if it hears other irrelevant information or does not receive any packet. All the witness nodes will send their tag value of $A \rightarrow B$ to node A after the verification ends, and node A will compute the sum denoted by s. Then, s and the monitoring result of node A itself will determine together whether there exist wormholes between node A and B.

If node A receives the reply from node B within time τ (τ is the maximum communication delay in WSNs), it will believe that there is no wormhole between node A and B, no matter what value s is. If node A monitors that the packet has been forwarded, it will think that a wormhole exists between node A and B, no matter what value s is. If node A does not hear any information within time τ, the judgment of whether there exist wormholes depends on the value of s. If $s \geq 1$, there is no wormhole. If $s < 1$, there exist a wormhole, node A and node B erase each other from their neighbor tables and notify their neighbors.

V. SIMULATION ANALYSIS

In this section, we present the evaluation results of CREDND. We first show our experimental setups and what parameters have an impact on CREDND. Then, we present our baseline methods and performance metrics. Finally, we present the evaluation and comparison result of CREDND as the parameters change.

A. EXPERIMENTAL SETUPS

This experiment is performed using MATLAB with programming in C language and the model uses the improved Group-based discovery protocol [20]. Each simulation repeated 20 times with confidence intervals of 95%, and the results were averaged. We deploy 800-1700 nodes randomly in a 1000×1000 square field. The maximum communication range of each node is $50m$ and the transmission range of the nodes varies from 32.5 m to 50 m to change the Degree of Irregularity (*DOI*) in order to make simulations closer to the real scene. After the communication of nodes is modeled, we arrange 2-10 wormholes in the network, with a half internal wormholes (real hops between two ends vary from 1 to 4) and a half external wormholes (real hops between two ends vary from 5 to 9). Finally, we test CREDND with *Wormhole threshold* varies from 3 to 7 and *Neighbor ratio threshold* varies from 1.1 to 1.5.

B. PARAMETER

We investigate the impact of some parameters on the performance of CREDND and chose the default values which make the system perform best. Those parameters are as follows:

1) *NodeDegree*: the degree of a node is the number of edges connected to the node, which can also be considered as the average node number per communication range of a node. The formula (4) shows how to calculate *NodeDegree*, where r_{max} is the maximum transmission range of sensor nodes, N is the total number of sensor nodes, L and W are the length and width of the network area respectively. When $N = 1000$, the *NodeDegree* is 7.85, that means there are about 8 nodes in the communication range of a node. *NodeDegree* ranges from 6.28 to 13.35 and the default value is 7.85.

$$NodeDegree = \pi r_{max}^2 \cdot \frac{N}{L \cdot W} \tag{4}$$

2) *DOI* (Degree of Irregularity): the irregularity of the signal radiation of the node, which is defined as the maximum change in signal intensity per unit angle along the direction of propagation of the signal. The communication between nodes may suffer varies interferences in the real world, so we studied the impact of *DOI* in order to make the simulation closer to the real scene. The upper boundary of *DOI* is r_{max}, and the lower boundary is (1-*DOI*)$\cdot r_{max}$. *DOI* ranges from 0 to 0.35 and the default value is 0.1.
3) *Wormhole number*: the number of wormholes in the network, it ranges from 2 to 10, and the default value is 4. Half of the wormholes are internal wormholes whose real hops vary from 1 to 4, and the others are external wormholes whose real hops vary from 5 to 9.
4) *Wormhole threshold*: we need to compare the hop count between two exclusive nodes with a threshold when

detecting external wormholes, which is called *wormhole threshold*. *Wormhole threshold* ranges from 3 to 7 and the default value is 4.

5) *Neighbor Ratio Threshold*: we proposed *Neighbor Ratio Threshold* to determine which nodes need wormhole detections. Suppose the neighbor number of a node with or without wormhole affected are n and $\bar{n}$, respectively. Then, the value of $n/\bar{n}$ is called neighbor ratio, which will be compared with *Neighbor Ratio Threshold*. *Neighbor Ratio Threshold* ranges from 1.1 to 1.5 in the simulation and the default value is 1.2.

C. BASELINE METHOD AND PERFORMANCE METRICS

In the experiments, we use the SECUND [45] and SEDINE [46] as the baseline, which also use the hop difference and local monitoring to detect wormholes. And the result without detecting is also our baseline. We use the following two performance metrics for performance evaluation: the number of bogus links and the rate of legal links not found.

D. EVALUATION AND COMPARISON

The following is the impacts of different parameters on CREDND.

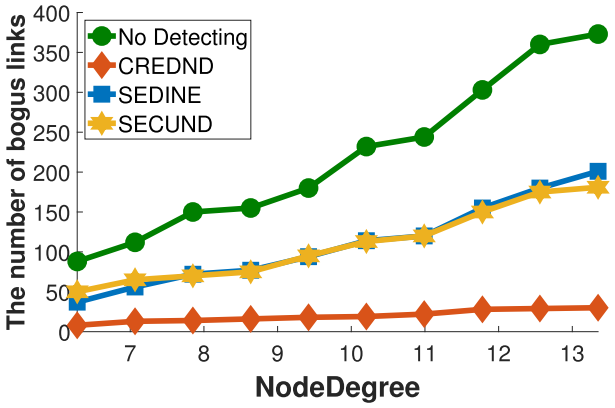

FIGURE 6. The impact of *NodeDegree* on different approaches.

1) THE IMPACT OF *NODEDEGREE*

From Fig.6 we can see that the number of bogus links of each approach increases as *NodeDegree* increases, for the reason that nodes located in the communication range of wormholes increase. SEDINE cannot effectively detect wormholes created by external malicious nodes using out-of-band. SECUND cannot detect wormholes with short links. So SECUND performs worse than SEDINE when *NodeDegree* is small because there are few nodes in each node's communication range where the hop counts between two ends of wormholes may be small. But SECUND perform better than SEDINE with the increase of *NodeDegree* as the actual hops of the wormhole getting bigger. CREDND can detect not only external wormholes but also internal wormholes, so it performs better than other algorithms. Besides, the number of bogus links of No Detecting sometimes increases rapidly and sometimes increases slowly, the reason

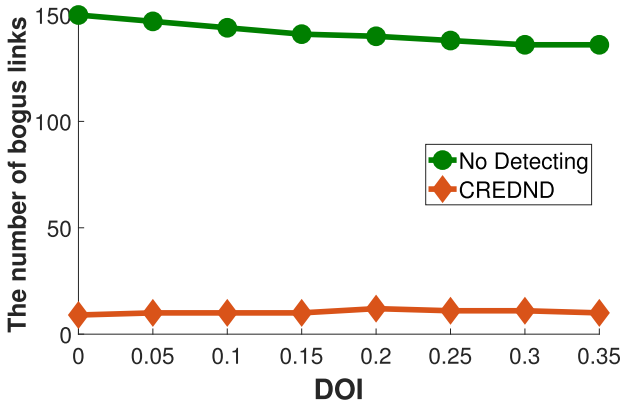

FIGURE 7. The impact of *DOI* on different approaches.

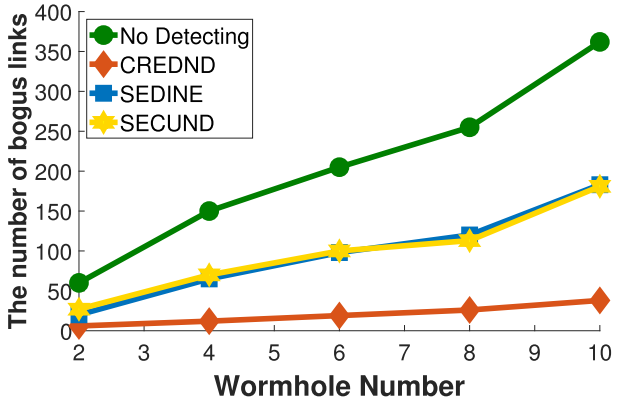

FIGURE 8. The impact of *wormhole number* on different approaches.

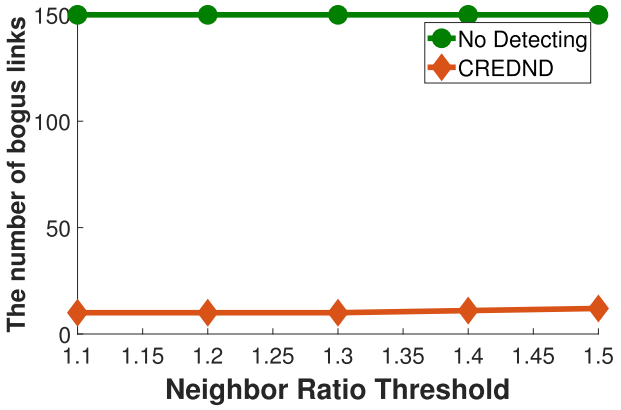

FIGURE 9. The impact of *neighbor ratio threshold* on different approaches.

may be that nodes are randomly deployed, and the newly deployed nodes may or may not locate in the communication range of wormholes.

2) THE IMPACT OF *DOI*

We only compare CREDND with No Detecting to better reflect the result of CREDND, because the difference between No Detecting and CREDND is the number of bogus links that CREDND discarded. From Fig.7 we can see that the number of bogus links of No Detecting decreases as *DOI* increases. The reason may be that a bigger *DOI* makes the radiation distance of nodes become shorter, and that cause

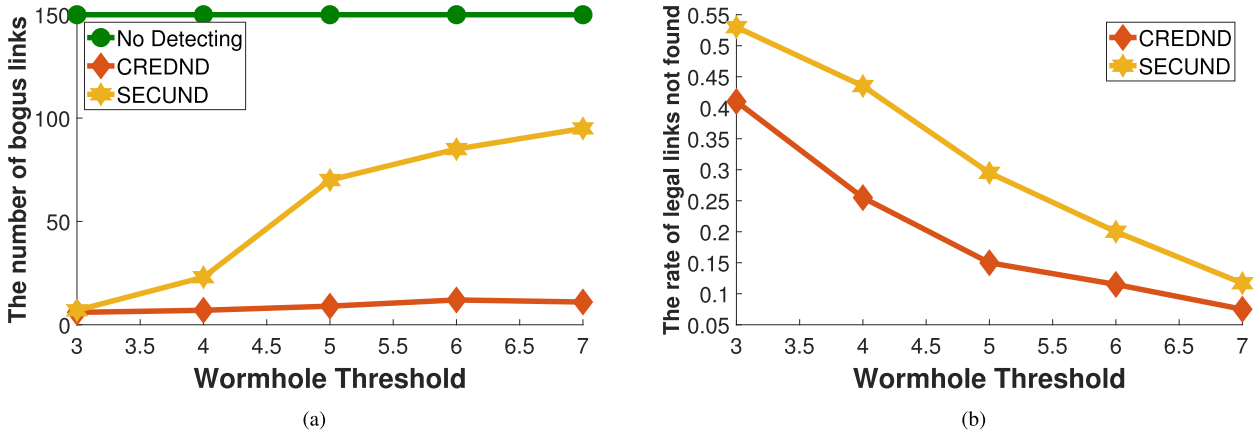

FIGURE 10. The impact of *wormhole threshold* in (a) the number of bogus links, (b) the rate of legal links not found.

some nodes located in the communication range of wormholes to go to out of the ranges. CREDND almost does not get any influence of *DOI*, as we ruled that two nodes are neighbor only if both nodes can receive the packets from each other.

3) THE IMPACT OF *WORMHOLE NUMBER*

Generally speaking, the more wormholes, the more bogus links. From Fig.8 we can see that the number of bogus links of each approach increases as *wormhole number* increases. The number of bogus links of CREDND increases slowly at the beginning and then increases a little faster when wormholes become more. The extreme case is that the whole network is affected by wormholes, which is hard for the majority of wormhole detection algorithms that are independent of extra hardware devices to detect wormholes successfully.

4) THE IMPACT OF *NEIGHBOR RATIO THRESHOLD*

We compare CREDND with only No Detecting, as *Neighbor Ratio Threshold* is proposed by our method. As Fig.9 shows, the bogus links of CREDND increases with the increase of *Neighbor Ratio Threshold*. That is because when *Neighbor Ratio Threshold* is larger, the standard of whether performing the wormhole detections of node pairs is higher and more wormholes can be left undetected. The wormhole detection will be more thoroughly when *Neighbor Ratio Threshold* is smaller.

5) THE IMPACT OF *WORMHOLE THRESHOLD*

Wormhole threshold has a great effect on detecting wormholes using hop difference, so we ignore SEDINE and compare CREDND with SECUND and No Detecting. From Fig.10(a) we can see that the number of bogus links of SECUND and CREDND increase as *wormhole threshold* increases. CREDND performs better than SECUND, especially when the *wormhole threshold* is higher than 4. The reason is that the higher *wormhole threshold* is, the more wormholes whose links which are relatively short cannot be detected by SECUND. But CREDND will perform the internal wormhole detections later for the short wormhole links. Besides, we can also find that the rate of legal links not found of SECUND and CREDND decrease as *wormhole threshold* increases from Fig.10(b). Therefore, the value of *wormhole threshold* should be considered according to the needs of scenarios in actual implementations.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose CREDND against wormholes in WSN, based on hop difference and local monitoring. CREDND can detect not only external wormholes but also internal wormholes. It improves the ability of wormhole defense in ND without additional hardware and saves node energy at the same time. We also propose the concept of the *Neighbor Ratio Threshold*, which contributes to improving the accuracy and energy efficiency of wormhole detection. Through the simulation experiment, we can conclude that CREDND has better performance in the wormhole detection than other same types of solutions.

However, there are still some shortcomings that we need to overcome in the future, such as CREDND cannot work well in the condition that all nodes in WSN with different communication range, dynamically changing and conforming to other distributions. The dependence on the *Neighbor Ratio Threshold* to determine where the detection needs to be performed also need to be reduced. It is of great significance to study wormhole detection under existing conditions (without any additional hardware devices), so we will seek more possible evasion techniques to ravel out these disadvantages in our future research, in order to apply CREDND to more complex conditions.

REFERENCES

[1] R. Deng, J. Chen, X. Cao, Y. Zhang, S. Maharjan, and S. Gjessing, "Sensing-performance tradeoff in cognitive radio enabled smart grid," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 302–310, Mar. 2013.

[2] X. Chen, Z. Zhao, and H. Zhang, "Stochastic power adaptation with multiagent reinforcement learning for cognitive wireless mesh networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2155–2166, Nov. 2013.

[3] S. Wang, Y. Wang, J. P. Coon, and A. Doufexi, "Energy-efficient spectrum sensing and access for cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 61, no. 2, pp. 906–912, Feb. 2012.

[4] Q. Luo and J. Wang, "FRUDP: A reliable data transport protocol for aeronautical ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 2, pp. 257–267, Feb. 2018.

[5] Q. Luo and J. Wang, "Multiple QoS parameters-based routing for civil aeronautical ad hoc networks," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 804–814, Jun. 2017.
[6] H. Chen, W. Lou, Z. Wang, J. Wu, Z. Wang, and A. Xia, "Securing DV-Hop localization against wormhole attacks in wireless sensor networks," *Pervasive Mobile Comput.*, vol. 16, pp. 22–35, Jan. 2015.
[7] J. Luo, D. Wu, C. Pan, and J. Zha, "Optimal energy strategy for node selection and data relay in WSN-based IoT," *Mobile Netw. Appl.*, vol. 20, no. 2, pp. 169–180, Apr. 2015.
[8] J. Chen, X. Cao, P. Cheng, Y. Xiao, and Y. Sun, "Distributed collaborative control for industrial automation with wireless sensor and actuator networks," *IEEE Trans. Ind. Electron.*, vol. 57, no. 12, pp. 4219–4230, Dec. 2010.
[9] Q. Yang, X. Zhu, H. Fu, and X. Che, "Survey of security technologies on wireless sensor networks," *J. Sensors*, vol. 2015, Dec. 2015, Art. no. 842392.
[10] B. Bhushan and G. Sahoo, "Recent advances in attacks, technical challenges, vulnerabilities and their countermeasures in wireless sensor networks," *Wireless Pers. Commun.*, vol. 98, no. 2, pp. 2037–2077, 2018.
[11] V. Teotia, S. K. Dhurandher, I. Woungang, and M. S. Obaidat, "Wormhole prevention using COTA mechanism in position based environment over MANETs," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 7036–7040.
[12] P. Kaur, D. Kaur, and R. Mahajan, "Simulation based comparative study of routing protocols under wormhole attack in manet," *Wireless Pers. Commun.*, vol. 96, no. 1, pp. 47–63, 2017.
[13] J. Padmanabhan and V. Manickavasagam, "Scalable and distributed detection analysis on wormhole links in wireless sensor networks for networked systems," *IEEE Access*, vol. 6, pp. 1753–1763, 2018.
[14] S. Geetha and V. C. Patil, "Graph-based energy supportive routing protocol to resist wormhole attack in mobile adhoc network," *Wireless Pers. Commun.*, vol. 97, no. 1, pp. 859–880, 2017.
[15] G. Kumar, M. K. Rai, and R. Saha, "Securing range free localization against wormhole attack using distance estimation and maximum likelihood estimation in Wireless Sensor Networks," *J. Netw. Comput. Appl.*, vol. 99, pp. 10–16, Dec. 2017.
[16] F. A. Khan, M. Imran, H. Abbas, and M. H. Durad, "A detection and prevention system against collaborative attacks in mobile ad hoc networks," *Future Gener. Comput. Syst.*, vol. 68, pp. 416–427, Mar. 2017.
[17] J. Vatn and T. Aven, "An approach to maintenance optimization where safety issues are important," *Rel. Eng. Syst. Saf.*, vol. 95, no. 1, pp. 58–63, 2010.
[18] S. Ji, T. Chen, and S. Zhong, "Wormhole attack detection algorithms in wireless network coding systems," *IEEE Trans. Mobile Comput.*, vol. 14, no. 3, pp. 660–674, Mar. 2015.
[19] L. Chen *et al.*, "Distributed range-free localisation algorithm for wireless sensor networks," *Electron. Lett.*, vol. 50, no. 12, pp. 894–896, Jun. 2014.
[20] L. Chen *et al.*, "Group-based discovery in low-duty-cycle mobile sensor networks," in *Proc. IEEE 9th Annu. Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw. (SECON)*, Jun. 2012, pp. 542–550.
[21] P. Lee, A. Clark, L. Bushnell, and R. Poovendran, "A passivity framework for modeling and mitigating wormhole attacks on networked control systems," *IEEE Trans. Autom. Control*, vol. 59, no. 12, pp. 3224–3237, Dec. 2013.
[22] S. Khurana and N. Gupta, "End-to-end protocol to secure *ad hoc* networks against wormhole attacks," *Secur. Commun. Netw.*, vol. 4, no. 9, pp. 994–1002, 2011.
[23] S. Jamali and V. Shaker, "Defense against SYN flooding attacks: A particle swarm optimization approach," *Comput. Elect. Eng.*, vol. 40, no. 6, pp. 2013–2025, 2014.
[24] M. Sookhak *et al.*, "Geographic wormhole detection in wireless sensor networks," *PLoS ONE*, vol. 10, no. 1, p. e0115324, 2015.
[25] T. T. Vo, N. T. Luong, and D. Hoang, "MLAMAN: A novel multilevel authentication model and protocol for preventing wormhole attack in mobile ad hoc network," in *Wireless Networks*. New York, NY, USA: Springer, 2018, pp. 1–18.
[26] C. Benzaid, K. Lounis, A. Al-Nemrat, N. Badache, and M. Alazab, "Fast authentication in wireless sensor networks," *Future Gener. Comput. Syst.*, vol. 55, pp. 362–375, Feb. 2016.
[27] J. von Mulert, I. Welch, and W. K. G. Seah, "Security threats and solutions in MANETs: A case study using AODV and SAODV," *J. Netw. Comput. Appl.*, vol. 35, no. 4, pp. 1249–1259, 2012.
[28] X. Wang and J. Wong, "An end-to-end detection of wormhole attack in wireless ad-hoc networks," in *Proc. IEEE 31st Annu. Int. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2007, pp. 39–48.
[29] Z. Shi, R. Sun, R. Lu, J. Qiao, J. Chen, and X. Shen, "A wormhole attack resistant neighbor discovery scheme with RDMA protocol for 60 GHz directional network," *IEEE Trans. Emerg. Topics Comput.*, vol. 1, no. 2, pp. 341–352, Dec. 2013.
[30] H. Chen, W. Chen, Z. Wang, Z. Wang, and Y. Li, "Mobile beacon based wormhole attackers detection and positioning in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 3, p. 910242, 2014.
[31] X.-W. Wang *et al.*, "Research on improved DV-HOP algorithm against wormhole attacks in WSN," in *Proc. 3rd Annu. Int. Conf. Inf. Technol. Appl. (ITA)*. Les Ulis, France: EDP Sciences, vol. 7, 2016, p. 03007.
[32] P. Kaur, D. Kaur, and R. Mahajan, "Wormhole attack detection technique in mobile ad hoc networks," *Wireless Pers. Commun.*, vol. 97, no. 2, pp. 2939–2950, 2017.
[33] S. Mukherjee, M. Chattopadhyay, S. Chattopadhyay, and P. Kar, "Wormhole detection based on ordinal MDS using RTT in wireless sensor network," *J. Comput. Netw. Commun.*, vol. 2016, Oct. 2016, Art. no. 3405264.
[34] P. Amish and V. B. Vaghela, "Detection and prevention of wormhole attack in wireless sensor network using AOMDV protocol," *Procedia Comput. Sci.*, vol. 79, pp. 700–707, 2016.
[35] J. Karlsson, L. S. Dooley, and G. Pulkkis, "Identifying time measurement tampering in the traversal time and hop count analysis (TTHCA) wormhole detection algorithm," *Sensors*, vol. 13, no. 5, pp. 6651–6668, 2013.
[36] F. Shi, D. Jin, W. Liu, and J. Song, "Time-based detection and location of wormhole attacks in wireless ad hoc networks," in *Proc. IEEE 10th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Nov. 2011, pp. 1721–1726.
[37] G. Wu, X. Chen, L. Yao, Y. Lee, and K. Yim, "An efficient wormhole attack detection method in wireless sensor networks," *Comput. Sci. Inf. Syst.*, vol. 11, no. 3, pp. 1127–1141, 2014.
[38] J.-W. Ho and M. Wright, "Distributed detection of sensor worms using sequential analysis and remote software attestations," *IEEE Access*, vol. 5, pp. 680–695, 2017.
[39] X. Lu, D. Dong, and X. Liao, "WormPlanar: Topological planarization based wormhole detection in wireless networks," in *Proc. 42nd Int. Conf. Parallel Process. (ICPP)*, Oct. 2013, pp. 498–503.
[40] G. Akilarasu and S. M. Shalinie, "Wormhole-free routing and DoS attack defense in wireless mesh networks," *Wireless Netw.*, vol. 23, no. 6, pp. 1709–1718, 2017.
[41] S. Jamali and R. Fotohi, "DAWA: Defending against wormhole attack in MANETs by using fuzzy logic and artificial immune system," *J. Supercomput.*, vol. 73, no. 12, pp. 5173–5196, 2017.
[42] D. S. K. Tiruvakadu and V. Pallapa, "Confirmation of wormhole attack in MANETs using honeypot," *Comput. Secur.*, vol. 76, pp. 32–49, Jul. 2018.
[43] S. Ji, T. Chen, S. Zhong, and S. Kak, "DAWN: Defending against wormhole attacks in wireless network coding systems," in *Proc. INFOCOM*, Apr./May 2014, pp. 664–672.
[44] T. Giannetsos and T. Dimitriou, "LDAC: A localized and decentralized algorithm for efficiently countering wormholes in mobile wireless networks," *J. Comput. Syst. Sci.*, vol. 80, no. 3, pp. 618–643, 2014.
[45] T. Hayajneh, P. Krishnamurthy, D. Tipper, and A. Le, "Secure neighborhood creation in wireless ad hoc networks using hop count discrepancies," *Mobile Netw. Appl.*, vol. 17, no. 3, pp. 415–430, 2012.
[46] S. Hariharan, N. B. Shroff, and S. Bagchi, "Secure neighbor discovery through overhearing in static multihop wireless networks," *Comput. Netw.*, vol. 55, no. 6, pp. 1229–1241, 2011.
[47] B. Liu and D. Towsley, "A study of the coverage of large-scale sensor networks," in *Proc. IEEE Int. Conf. Mobile Ad-Hoc Sensor Syst.*, Oct. 2004, pp. 475–483.

XIAO LUO is currently the Director of the Second Research Institute of Civil Aviation Administration of China. His research interests include electronic information of civil aviation airport, logistics, and aviation security. As a Technical Leader and the Manager of aviation, he has received many national honors and has published many academic papers. He is also a Committee Member of the 11th All-China Youth Federation and a Standing Director of the China Association for Science and Technology.

YANRU CHEN received the master's degree in management (finance) from the University of Melbourne, Australia. She is currently pursuing the Ph.D. degree with the Embedded Systems and Big Data Management Laboratory and the Internet of Things Laboratory, School of Computer Science, Sichuan University, under the supervision of Prof. B. Guo. Her research interests include wireless sensor networks, cognitive psychology, personal financial big data, computational finance, and financial data space.

MIAO LI received the M.S. degree from the School of Computer Science, Sichuan University, under the supervision of Prof. L. Chen. His research interest includes the Internet of Things.

QIAN LUO received the Ph.D. degree from Sichuan University, Chengdu, China. He is currently with the Second Research Institute of Civil Aviation Administration of China. He is also an expert in the field of information technology. He is also a Researcher. He has accomplished several key national projects in aviation area and has published many academic papers.

KANG XUE was born in Chengdu, China. He received the B.Sc. degree from Sichuan University, Chengdu, in 2003. He is currently an Associate Professor with the Second Research Institute of Civil Aviation Administration of China. His research interests include civil aviation communication, navigation, and surveillance technology.

SHIJIA LIU is currently pursuing the Ph.D. degree with the School of Computer Science, Sichuan University, under the supervision of Prof. L. Chen. Her research interest includes Wi-Fi-based activity recognition.

LIANGYIN CHEN received the Ph.D. degree from the School of Computer Science, Sichuan University, in 2008, where he is currently a Professor. From 2009 to 2010, he was a Visiting Researcher with the University of Minnesota, under the supervision of Prof. T. He. He has authored or co-authored more than 40 papers, many of which were published in premier network journals and conferences. His research interests include wireless sensor networks, embedded systems, computer networks, distributed systems, big data analytics, natural language processing, the Internet of Things, and Industrial Internet.