frontiers | Frontiers in Energy Research

TYPE Original Research
PUBLISHED 27 September 2022
DOI 10.3389/fenrg.2022.1017932

OPEN ACCESS

EDITED BY
Chunhe Song,
Shenyang Institute of Automation (CAS), China

REVIEWED BY
Xinli Wang,
Shandong University, China
Wenhuan Wang,
Zhejiang University of Technology, China

*CORRESPONDENCE
Xue Zhou,
xuezhouapplication@126.com

SPECIALTY SECTION
This article was submitted to Smart Grids, a section of the journal Frontiers in Energy Research

RECEIVED 12 August 2022
ACCEPTED 29 August 2022
PUBLISHED 27 September 2022

CITATION
Liu W, Chen Z, Yu X and Zhou X (2022), A cluster-based approach against wormhole attacks in MANETs among smart grid. *Front. Energy Res.* 10:1017932.
doi: 10.3389/fenrg.2022.1017932

A cluster-based approach against wormhole attacks in MANETs among smart grid

Weijie Liu, Zhiran Chen, Xiang Yu and Xue Zhou*

Research Department of Industrial Development, Zhejiang Development and Planning Institute, Hangzhou, China

Mobile ad hoc networks in smart grid become more and more popular and significant. However, the deployment scenarios, the functionality requirements and the limited capabilities make them vulnerable to a large group of attacks, e.g., wormhole attacks. In this paper, a novel cluster-based scheme is proposed for the purpose of preventing wormhole attacks. Firstly, a clustering algorithm is proposed that employs a powerful analytical hierarchy process methodology to elect clusterheads. Afterwards, the elected clusterheads are required to implement the wormhole attacks prevention scheme which includes two phases, i.e., detection phase and location phase. By detection phase, the existence of wormhole attacks can be detected. By location phase, the wormhole nodes are able to be detected. Simulation results indicate the scheme in our paper can be used to prevent wormhole attacks in ad hoc networks efficiently.

KEYWORDS

analytical hierarchy process, cluster, smart grid, wormhole, AHP

1 Introduction

Mobile Ad Hoc Network (MANET) is a hot research area delivering Intelligent Transportation System (ITS) services to end users. Routing in MANETs has been frequently studied and discussed over the past years. MANETs have several merits, such as providing drivers with frequent and timely road conditions information, reducing road accidents, etc. The ultimate goal of MANETs is to efficiently transfer information between mobiles. If an unauthorized user tampers with the message, or causes a change in the network topology, the transmitted message can have a large impact on the driver's behavior. In MANET, attackers create problems by launching a large number of attacks to disrupt network conditions and threaten security, such as sending false information, forging data, disrupting traffic, violating privacy, etc. Therefore, security is extremely important in MANETs and also a major challenge. Malicious messages may directly or indirectly damage people's property and life in MANETs.

MANET belongs to Internet of Things (IoT) networks in which edge computing are among moble nodes. Therefore, in compared with the other networks, MANETs are more vulnerable to be attacked because of their unique characteristics (Adarsh et al., 2021). In this paper, the prevention of wormhole attacks is our focus. In wormhole attacks, a

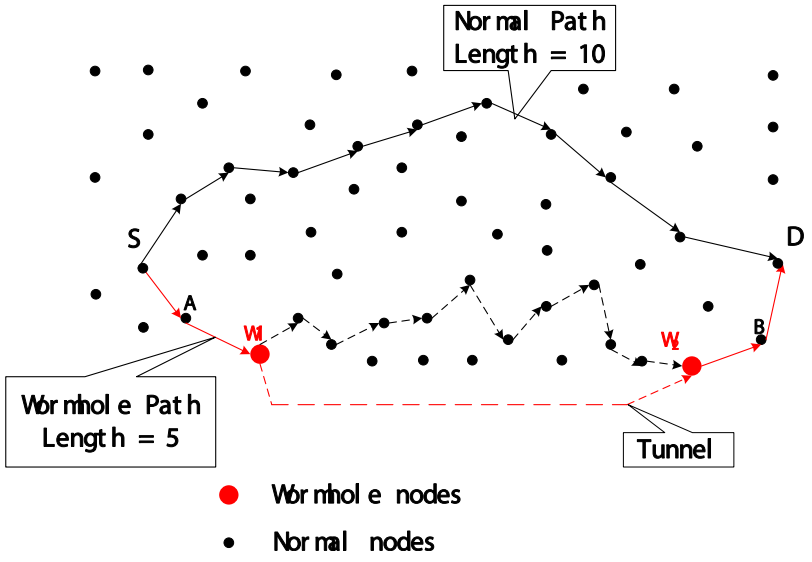

FIGURE 1
Scenario of wormhole attack.

wormhole comprises with two colluding malicious nodes, named wormhole nodes that are far from each other, and a tunnel between them. The tunnel between wormhole nodes can be a dedicated communication medium such as long range wireless devices or optical cable (Ghugar and Pradhan, 2021). One wormhole node captures routing traffic at one point of the network and tunnels them to its peer wormhole node at another point of the network. Hence, the network topology is corrupted and routing is compromised. Afterward, wormhole nodes could exploit the data in variety of ways: selectively dropping packets to interrupt communication, trying to crack communication keys, etc. Because wormhole nodes need not to modify or create new packets, no cryptographic technique can prevent MANETs from wormhole attacks (Tahboush and Agoyi, 2021). In this type of attack, malicious nodes overhear messages transferred over the communication channel that may be used by serious threats (Afzal and Kumar, 1427).

As shown in **Figure 1**, a wormhole node (W_1) encapsulates packets and sends them to its peer wormhole node (W_2) through the path between them. After that, the wormhole node W_2 can get the original data packet extracted from the encapsulated data packet, named decapsulation (Ghugar and Pradhan, 2021). This process of encapsulation and decapsulation is called a tunnel, and the path between peer nodes is a tunnel. Because the original packets are encapsulated, they are not altered by intermediate nodes along the path between W_1 and W_2. So it seems that W_2 gets packets directly from W_1 with the same number of hops, although they are usually many hops away from each other. Therefore, paths with wormhole nodes may be shorter than other normal paths. Therefore, compared with other normal paths, most senders prefer to choose the path with the wormhole node as the routing path to transmit packets. For example, in **Figure 1**, the path from source S and destination D in between wormhole W_1 and W_2 is five hops long, while the normal path is 10 hops long. Therefore, for all routing protocols, for example Ad hoc On Demand Distance Vector (AODV) protocol (Sarhan and Sarhan, 2021) and Dynamic Source Routing (DSR) protocol (Tiado et al., 2021), node S prefers to send packets to node D along the path Wormhole W_1 and W_2.

In this paper, a new cluster-based scheme is proposed to prevent wormhole attacks in MANET. The structure of our paper is as below. The next section introduces related works. In **Section 3**, our new clustering algorithm is proposed to select cluster heads (CHs) that perform wormhole attack prevention. **Section 4** introduces a detection method in detecting the presence of wormhole attacks in the network. Subsequently, **Section 5** proposes a localization mechanism to identify wormhole nodes. **Section 6** gives performance simulations. At last, our paper is summarized briefly in **Section 7**.

2 Related works

In (Garg et al., 2019), the authors introduce a wormhole attack detection system using machine learning which

determines the behavior of mobiles in MANETs. This system uses the trace files produced by the simulator which consists of both normal and abnormal behavior of nodes. The obvious weakness of the method in this paper is the hug time cost to generate a set of data that will be used to learn the attack. The method based on Artificial Neural Network machining learning is not suitable for detecting wormhole attacks in the environment of MANET especially considering the hug collection cost of transmitted packets and training.

In (Krundyshev et al., 2018), the authors propose the approach to provide security for MANET and other types of transport relative networks using swarm algorithms of artificial intelligence. The proposed swarm algorithm is used to locate wormhole attack in MANETs. The algorithm in this paper uses Intelligent Water Drops (IWD) with trust model. Trust is the basic element in making a trusted mobile environment which promotes security in mobile networks. Trust value of neighbor node is calculated. Average value of the pheromone per unit calculates and then checks to the threshold value if exceeded node secure either not secure. The limitation of this paper is that parameters are re-initialized after each iteration of the IWD algorithm which leads to high cost in MANET.

In (Ahutu and El-Ocla, 2020), the authors describe a secure routing protocol using the lightweight multi-hop routing protocol called MAC Centralized Routing Protocol (MCRP). It uses concept of the time ratio threshold to locate wormhole attacks and malicious nodes. The authors assume that normal nodes are only responsible for data forwarding. The key point in the MCRP routing protocol is the implementation of centralized network intelligence to reduce the energy consumption while maintaining the consensus between nodes. Accordingly, it can detect wormhole attacks. However, some nodes may be incorrectly identified as wormhole nodes because they are located in prime locations for connections within the network.

To avoid the weaknesses of the above proposals, we propose our wormhole attack prevention scheme.

3 Preliminaries

Our mechanism is outlined below. First of all, the whole network is cutted into clusters. CHs are elected in corresponding clusters. Then, CH needs to perform a wormhole attack prevention scheme including a detection phase and a localization phase in each cluster. Therefore, three questions should be addressed (Adarsh et al., 2021): How to elect a CH for each community? (Ghugar and Pradhan, 2021)? How does CH detect the existence of a wormhole attack? (Tahboush and Agoyi, 2021)? How does CH identify wormhole nodes?

3.1 Preparatory work: Three parameters

First, to answer the first question mentioned above, a new clustering algorithm is proposed that uses the Analytic Hierarchy Process (AHP) (Raghav et al., 2022) to calculate the weight of every node in MANETs. The node having the largest nearby weight value is selected as the CH in the corresponding cluster. In this subsection, three parameters will be introduced to jointly determine the weight of each node, including relative stability (S_r), connectivity value (C_v), and forward exchange rate reciprocal (R_f). The first parameter S_r is evaluated in terms of the rate of change of the neighbors $'$. The second parameter C_v is evaluated based on the value of distance to neighbors and the number of neighbors, also known as distance-considered connectivity. The third parameter R_f is evaluated based on the packet forwarding rate. In short, the nodes remain relatively stable or move slowly, the closer the neighbors, the lower the forwarding rate and the greater the chance of being selected as the CH. Conversely, a node with a fast moving speed, or a node with fewer and fewer neighbors, or a node with a higher forwarding rate, has a smaller chance of becoming a CH in the corresponding vicinity. The function of each parameter and the evaluation process are described below.

3.1.1 Relative stability (S_r)

The need for relative stability (S_r) needs to be explained. Consider the following scenarios (Adarsh et al., 2021): A node moves so fast compared to its neighbors that its connections and communications with other nodes are very brief. In this case, this node is of little use and should not be assigned important responsibilities (Ghugar and Pradhan, 2021). If a node with relatively fast mobility compared with its neighbors is selected as the CH, the cluster managed by this node will collapse quickly. Therefore, cluster reassociation must be performed, which greatly increases the overhead. Considering these situations, our scheme is to choose a relatively stable CH that can stay nearby for a longer time, rather than a node with high mobility. Therefore, a relative stability parameter is involved to compute the stability of each node.

The value of S_r for a node is evaluated based on changes in its neighborhood. As the parameter name implies, it is a relative value. Relative stability is defined as the stability compared to hop-on neighbors. In contrast to relative stability, absolute stability is the stability compared to some frame of reference such as a laboratory floor or road. Relative stability is more useful in terms of data transfer rates, given the frequent changes in topology in MANETs. It indicates whether a node is moving relatively fast or slow or even steady in its neighbors compared to its neighbors.

Our method uses graph theory (He et al., 2022) and similarity calculation method (Hu et al., 2022) to calculate the relative stability of every node. A network with nodes and links

is a directed graph, $G(t) = (V(t), E(t))$, which is called a neighbor relationship graph, where $V(t) = \{v_1, v_2, \ldots, v_n\}$ represents the set of participating nodes, $E(t) = \{e_1, e_2, \ldots, e_m\}$ represents the set of wireless links. If v_i gets the data from v_j, there is a directed edge $e(i,j)$ between v_i and v_j. It means that v_j is the neighbor of v_i.

$V_i(t_j)$ and $V_i(t_{j+1})$ are vectors, representing node v_i within the transmission area of some consecutive two time points t_j and $t_{j+1Mobilenodesituation}$. $E_i(t_j)$ and $E_i(t_{j+1})$ represent the wireless link situation of node v_i in the transmission area of some two consecutive time points t_j and $t_{j+vector1}$. According to similarity theory (Jaccard index), the stability of nodes in the transmission region of node v_i can be calculated by the average similarity value between $V_i(t_j)$ and $V_i(t_{j+1})$ as shown in **Eq. (1)**.

$$S_{node}\left(t_j, t_{j+1}\right) = \frac{1}{P-1}\sum_{j=1}^{P-1}\cos\theta_j = \frac{1}{P-1}\sum_{j=1}^{P-1}\frac{V_i\left(t_j\right)\cap V_i\left(t_{j+1}\right)}{V_i\left(t_j\right)\cup V_i\left(t_{j+1}\right)} \quad (1)$$

where θ_j is the angle between $V_i(t_j)$ and $V_i(t_{j+1})$, and P is the number of time points at which $V_i(t_j)$ was observed. $S_{node}(t_j, t_{j+1})$ represents the similarity between the state of the neighboring situation of node v_i's at different time points, such as t_j and t_{j+1}. If S_{node} is larger, the angle between $V_i(t_j)$ and $V_i(t_{j+1})$, i.e. $\hat{I}z_j$, will be smaller, which means The similarity between $V_i(t_j)$ and $V_i(t_{j+1})$, that is, the neighbors of node v_i at the time points of t_j and t_{j+1} will not be Dynamic changes.

Similarly, the link stability between nodes in the transmission area of v_i is computed by the average similarity value between $E_i(t_j)$ and $E_i(t_{j+1})$ as shown in **Eq. 2**.

$$S_{link}\left(t_j, t_{j+1}\right) = \frac{1}{P-1}\sum_{k=1}^{P-1}\cos\theta_k = \frac{1}{P-1}\sum_{j=1}^{P-1}\frac{E_i\left(t_j\right)\cap E_i\left(t_{j+1}\right)}{E_i\left(t_j\right)\cup E_i\left(t_{j+1}\right)} \quad (2)$$

where θ_k is the angle between the two vectors $E_i(t_j)$ and $E_i(t_{j+1})$, and P is the number of time points at which $E_i(t_j)$ is observed. $S_{link}(t_j, t_{j+1})$ represents the similarity of different link vectors in the transmission area of node v_i at different time points, such as t_j and t_{j+1}. If S_{link} is larger, the angle between $E_i(t_j)$ and $E_i(t_{j+1})$, ie θ_k will be smaller, which means $E_i(t_j)$ and $E_i(t_{j+1})$, that is, the link within the transmission range of v_i does not change dynamically at the time points of t_j and t_{j+1}.

At last, based on the similarity theory, relative stability of a node is computed using S_{node} and S_{link} as shown in **Eq. 3**.

$$S_r = \alpha \cdot S_{node} + (1-\alpha)\cdot S_{link} \quad (3)$$

where α is the weighting factor evaluated by **Eq. 4**.

$$\alpha = \frac{V_i\left(t_j\right)\cap V_i\left(t_{j+1}\right)}{V_i\left(t_j\right)\cap V_i\left(t_{j+1}\right) + E_i\left(t_j\right)\cap E_i\left(t_{j+1}\right)} \quad (4)$$

The rationality of the setting of α is analyzed as follows.

- First of all, the filling-in of α does not change the property of the function of S_r in terms of increasing function or decreasing function, e.g., S_r is still a increasing function as the intersection of $V_i(t_j)$ and $V_i(t_{j+1})$ or the intersection of $E_i(t_j)$ and $E_i(t_{j+1})$ increases.
- α is always larger than $1-\alpha$, so the effect of S_{node} is enhanced purposely by the weighting in Equation. This is due to the consideration that the variation of neighbors is more important than that of links, since link is more frangible in MANETs.

The main feature of MANET is its dynamic topology, where nodes change positions randomly. CH should change as little as possible while moving. Therefore, a node that moves slowly compared to its neighbors is chosen as the CH; otherwise, the cluster may become corrupted. Therefore, nodes with larger S_r are better suited to play the CH role.

An illustrative case is as below. In **Figure 2**, two groups of nodes at time points t_j and t_{j+1} are $V_i(t_j) = \{v_a, v_b, v_c, v_d, v_e\}$ and $V_i(t_{j+1}) = \{v_a, v_b, v_c, v_e, v_f\}$; the corresponding two sets of links are $E_i(t_j) = \{e_{ab}, e_{bc}, e_{cd}, e_{de}\}$ and $E_i(t_{j+1}) = \{e_{ab}, e_{ac}\}$, respectively.

3.1.2 Connectivity value (C_v)

The second parameter is computed using the number of node′s neighbors and their distance values from different neighbors, named the connectivity value (C_v), as shown in **Eq. 5**.

$$C_v = \frac{C(i)}{D(i)} \quad (5)$$

where C(i) is the connectivity degree that stands for the neighbor counts of v_i. D(i) is the distance value sum between v_i and its neighbors calculated by **Eq. 6**.

$$D(i) = \sum_{j=1, j\neq i}^{N}\sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \quad (6)$$

where N is the number of nodes in V that stands for the neighbors of v_i. Node i is a neighbor of node j within the transmission range of j. (x_i, y_i) and (x_j, y_j) are the coordinates of v_i and v_j respectively.

If a node has more neighbors, the node is in a more important position. Therefore, nodes with greater connectivity should be more likely to be selected as CHs. If a CH has distant members, more power is needed when nodes are far away. Therefore, nodes with a smaller sum of distances are more popular than nodes with a larger sum of distances. Nodes with a big number of neighbors and a small distance from their neighbors have a higher chance to be elected as a CH, which can minimize node separation and enhance the stability of the cluster. **Figure 3** shows an illustrative example.

3.1.3 Reciprocal of forward rate (R_f)

R_f is evaluated based on the packet forwarding rate, as shown in **Eq. 7**. It indicates whether the node violates the backoff mechanism (?) specified by DCF in 802.11 so that they have

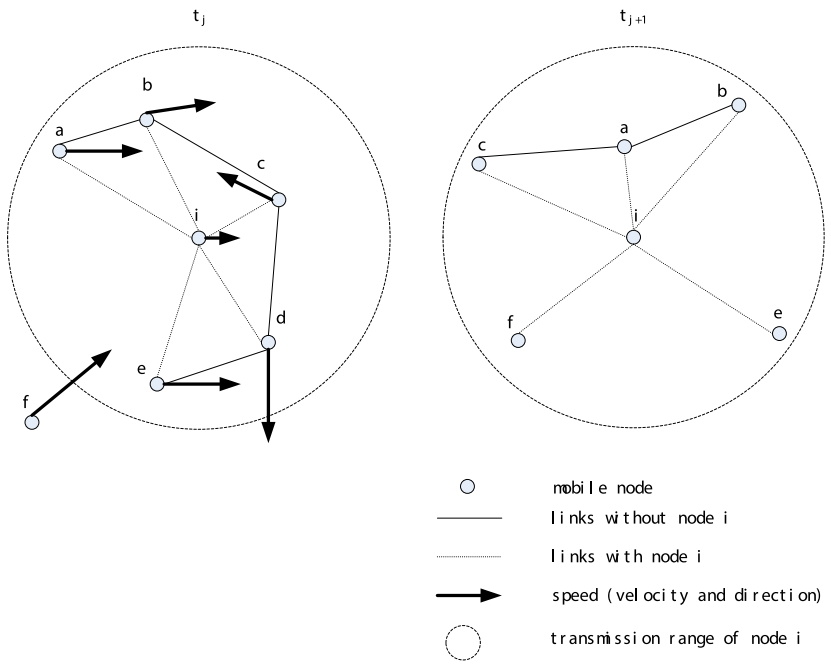

FIGURE 2
Illustration for relative stability.

a higher chance of blocking the wireless channel, or the node performs a wormhole attack, reducing the number of hops of the routing path to capture passing packets. It stands for the remaining battery power since frequent packet forwarding enhances the battery drain rate.

$$R_f = \frac{1}{F_r} = \frac{1}{N_j^{act}/t_c} = \frac{t_c}{N_j^{out} - N_j^{src}} \tag{7}$$

where F_r represents the forward exchange rate. N_j^{act} represents the number of packets forwarded by j. t_c is the time spent collecting evidence. In other words, N_j^{act} packets were observed during t_c. N_j^{out} represents the number of packets of "com out" of node j. N_j^{src} refers to the number of packets originating from node j. In short, when calculating the parameter R_f, only the forwarded packets are involved, not all transmitted packets, i.e., the packets generated by node j are not considered in the calculation of R_f.

3.2 Election of clusterhead

The calculation of the three parameters S_r, C_d and R_f is introduced. The weight of each node will then be calculated by considering these three parameters. To make CH election decisions in an organized way, the AHP method can be used to decompose the decision into the following four steps to generate priorities.

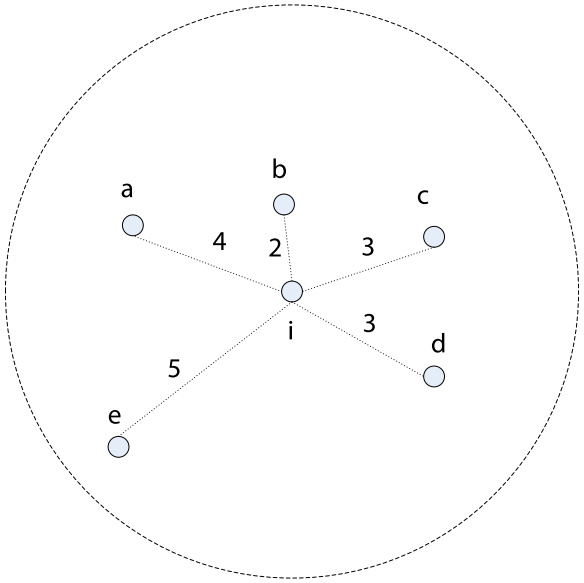

FIGURE 3
Illustration for connectivity value.

- Step 1: Define the issue.

The decision issue is to elect an appropriate node by considering the corresponding weight value as the cluster head in one-hop neighborhood in the MANETs.

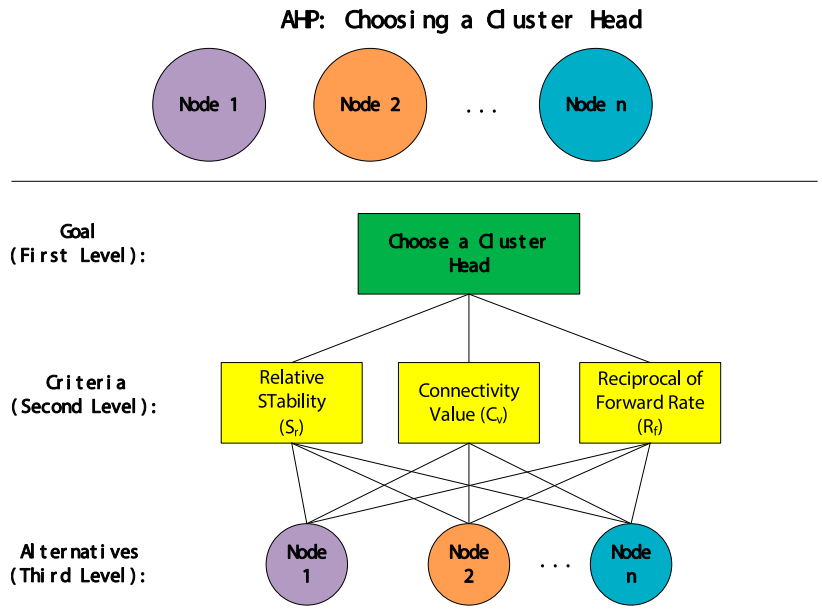

FIGURE 4
Hierarchical structure.

- Step 2: Build a decision hierarchy from top to the goal, afterwards from the broad perspective, through middle to the lowest level. The middle level refers the criteria subsequent elements depend on. The lowest level refers a set of alternatives

Figure 4 shows structuring the problem into a hierarchy. The overall goal of selecting a suitable CH is at the top of the hierarchy. Subsequent levels representing primary criteria are called secondary objectives. The three secondary goals are S_r, C_v and R_f. At last, alternatives are put at the bottom of the hierarchy to evaluate CH election.

- Step 3: Make a group of pairwise comparison matrices.

For comparison, it requires a numerical scale that indicates how many times more dominant one element than another on the criterion. Table 1 refers the scale.

The reciprocal matrix is constructed by comparing each criterion pairwise with another criterion present under the

TABLE 1 Fundamental scale of absolute numbers.

Intensity of importance	Definition	Explanation
1	Equally important	2 Activities also serve purpose
2	Weak or Slight	Experience and judgment support one activity slightly more than another
3	Moderate importance, experience and judgment reasonably support one activity over another 4	Moderate Plus
Experience and Judgment Moderately support one activity over another	—	—
5	Strong importance, experience and judgment strongly favor one activity over another 6	Strong Plus
Experience and judgment support one activity more strongly than another 7	very strong	events are recognized more strongly than others
8	very very strong	some activities are supported more strongly than others. Its superiority has actually been proven
9	Very Important	Evidence in favor of one activity over another is the highest positive order possible

highest target. The values of the pairwise comparison matrix are provided by answering questions that gain more preference and the degree of preference. The criteria matrix A gives pairwise comparisons of the three criteria against the highest target, as shown below. The value a_{ij} represents the preference strength of the *ith* criterion over the *jth* criterion. In Table 1, a basic one to nine scale is used to express the strength of preference based on experience and knowledge. The A matrix is explained in Eq. 8 below.

$$A = \left(a_{ij}\right) = \begin{pmatrix} S_r \\ C_v \\ R_f \end{pmatrix} \begin{pmatrix} S_r & C_v & R_f \end{pmatrix} = \begin{pmatrix} 1 & a_{S_rC_v} & a_{S_rR_f} \\ \frac{1}{a_{S_rC_v}} & 1 & a_{C_vR_f} \\ \frac{1}{a_{S_rR_f}} & \frac{1}{a_{C_vR_f}} & 1 \end{pmatrix} \tag{8}$$

A can be normalized to a normalized vector matrix A^{norm} by using mean normalization of row vectors, as shown in Eq. 9.

$$A^{norm} = \left(\frac{a_{ij}}{\sum_{i=1}^{k} a_{ij}}\right) \tag{9}$$

where k is the number of criteria. In our mechanism, three parameters are named the criteria to evaluate weight value of every node, so k is equal to 3.

Then, by normalizing the mean of the row vector, the normalized vector W_i^T can be got shown in Eq. 10, representing the weight factor of each criterion.

$$W_i^T = \left(w_j\right) = \left(\frac{1}{k}\sum_{j=1}^{k}\left(\frac{a_{ij}}{\sum_{i=1}^{k} a_{ij}}\right)\right) \tag{10}$$

Check all pairwise comparison matrices for consistency. Since people′s random judgment matrix can be prone to judgment errors, such judgment errors can be detected by the consistency ratio (CR) that is named as the ratio of consistency index (CI) to random index (RI). CI is calculated by Eq. (11) using standard matrix C as a instance. Consistency is considered after computing the weights for each criterion.

$$CI = \frac{\lambda - n}{n - 1} \tag{11}$$

where n refers the element counts compared in criteria matrix A, here it is 3. λ is evaluated by Eq. 12.

$$\lambda = \frac{\sum_{i=1}^{n} \mu_i}{n} \tag{12}$$

where μ_i is the consistency vector that is computed by Eq. 13.

$$\mu_i = \frac{\sum_{j=1}^{n} w_j a_{ij}}{w_i} \tag{13}$$

where w_i is the weight factoring of each criterion calculated by the aforementioned Eq. 10.

TABLE 2 Random index.

Exponent Number	1	2	3	4	5	6
RI	0	0	0.58	0.90	1.12	1.24

Finally, the consistency ratio (CR), the ratio between CI and RI, can be obtained as shown in Eq. 14. RI as shown in Table 2.

$$CR = \frac{CI}{RI} \tag{14}$$

When CR<0.1, the consistency of matrix is extremely acceptable that means the judgment error is tolerable. Otherwise, the pair wise matrix will be adjusted until the matrix satisfies the consistency check.

- Step 4: Use the priorities made by the comparisons to weigh the priorities in the next level. Repeat it for each element. Then, for every element in level below increase its weighed values and get its global priority. Repeat this weighing process until the final priorities of the alternatives in the bottom most level are completed.

Calculate local weights for every criterion and alternative. After computing the weight of every criterion, the weight of each node should be computed in the same way. It needs to compare nearby nodes from every angle of every criterion. Therefore, the following three matrices can be obtained, namely A_{S_r}, A_{C_v} and A_{R_f}, which are the corresponding local weight factor parameters, such as Eqs. 15–17 are shown. A_{S_r} represents pairwise comparison of nearby nodes according to criterion S_r. A_{C_v} represents a pairwise comparison of nearby nodes according to the criterion C_v. A_{R_f} represents a pairwise comparison of nearby nodes according to the criterion R_f.

$$A_{S_r} = \left(a_{ij}^{S_r}\right) = \begin{pmatrix} a_{n_1n_1}^{S_r} & a_{n_1n_2}^{S_r} & a_{n_1n_3}^{S_r} \\ \frac{1}{a_{n_1n_2}^{S_r}} & a_{n_2n_2}^{S_r} & a_{n_2n_3}^{S_r} \\ \frac{1}{a_{n_1n_3}^{S_r}} & \frac{1}{a_{n_2n_3}^{S_r}} & a_{n_3n_3}^{S_r} \end{pmatrix} \tag{15}$$

$$A_{C_v} = \left(a_{ij}^{C_v}\right) = \begin{pmatrix} a_{n_1n_1}^{C_v} & a_{n_1n_2}^{C_v} & a_{n_1n_3}^{C_v} \\ \frac{1}{a_{n_1n_2}^{C_v}} & a_{n_2n_2}^{C_v} & a_{n_2n_3}^{C_v} \\ \frac{1}{a_{n_1n_3}^{C_v}} & \frac{1}{a_{n_2n_3}^{C_v}} & a_{n_3n_3}^{C_v} \end{pmatrix} \tag{16}$$

$$A_{R_f} = \left(a_{ij}^{R_f}\right) = \begin{pmatrix} a_{n_1n_1}^{R_f} & a_{n_1n_2}^{R_f} & a_{n_1n_3}^{R_f} \\ \frac{1}{a_{n_1n_2}^{R_f}} & a_{n_2n_2}^{R_f} & a_{n_2n_3}^{R_f} \\ \frac{1}{a_{n_1n_3}^{R_f}} & \frac{1}{a_{n_2n_3}^{R_f}} & a_{n_3n_3}^{R_f} \end{pmatrix} \tag{17}$$

4 Detection phase for detecting wormhole attacks

Through the clustering algorithm mentioned earlier, MANET is divided into several clusters, as shown in **Figure 5**. An overlay is took in this paper tha is a virtual layer composed by CHs and gateways (GWs). CH needs to be responsible for the corresponding cluster and implement our wormhole attack prevention scheme. In the figure, the corresponding CHs are marked as CH_1, CH_2, CH_3 and CH_4 respectively. The role of gateways is to connect clusters and ensure data transmission between clusters, and can be divided into conventional gateways (RGW) and distributed gateways (DGW). An RGW is defined as a node in the overlapping region of two adjacent clusters, such as node RGW_1. DGWs reside in two adjacent clusters respectively and can communicate directly with each other, such as nodes DGW_1 and DGW_2.

Similar to AODV, the RREQ is broadcasted by source S to initiate the routing request process. Compared to AODV which uses a 2-tuple (source node IP, broadcast ID) to identify whether a RREQ is duplicated, in our scheme, a 3-tuple (source node IP, broadcast ID, first node IP) is used. The first node is named as a node located in the one-hop neighborhood of the source, sya the number of hops between the first node and the source is one. Triplets bring several benefits. One is to mitigate broadcast storms caused by duplicate RREQs, since intermediate nodes do not handle duplicate RREQs that go through the same first node. Another benefit brought by the first node mechanism is that multiple loop-free paths can be maintained, which can reduce the frequency of restarting the routing request process.

The process of node receiving RREQ is shown in **Figure 6**. Duplicate RREQs are simply discarded. The node that receives the RREQ will perform a process from the following two candidate actions, depending on if it is a target or an intermediate node.

- If the node decides that it is an intermediate node, i.e. not the destination node, it checks the hop count extracted from the RREQ. If the RREQ is greater than the number of hops in the corresponding entry in the routing table, RREQ is dropped. Or else it puts in a new entry in the routing table to make multiple routing paths. In short, all duplicate RREQs are discarded directly, and RREQs with smaller hop counts are used for intermediate nodes to create reverse paths for the following RREPs.
- If the node decides that it is the destination, it verifies if the sequence number of RREQ given by the source is bigger than the sequence number of any entry in the routing table. Or else, the destination directly drops the RREQ; if it is, it means that the RREQ has expired, and the destination node will reply RREP for each RREQ on the reverse routing path. No matter how many RREQs the destination node receives, it will reply RREPs to those corresponding RREQs, unless the sequence number of RREQ is smaller than that in the destination node′s routing table. In **Figure 7**, when RREP reaches the source node, it maintains multiple routing paths.

In our scheme, it is regulated that only the destination node has the permission to reply RREP to RREQ. Compared with AODV, where any intermediate node that owns the path to the destination node can reply RREQ with RREP, in our scheme, it is forbidden. In AODV case, the detection of wormhole attacks cannot be guaranteed, since wormhole nodes may not be involved in the routing path with the intermediate node that reply the RREP. Conversely, in our scheme, all nodes including wormhole nodes are required to participate into the routing process in order to expose the wormhole nodes' existence. In routing request process, every intermediate node lists the hop count to the source when the RREQ passes through the routing path, denotes as Hop_{RREQ}. On the other hand, in routing reply process, each intermediate node records its hop count value to the destination node when the RREP passes through the reversed routing path, denotes as Hop_{RREP} as shown in **Figure 7**.

As shown in **Figure 6**, the RREQ retransmitted by node five is dropped by node 4, since the RREQ′s first node is node one which is the same with the RREQ received by node four from node 1. Thus, the broadcast storm caused by the potential loop among node 1, node four and node five is avoided. As it is assumed that the node W_1 and W_2 are wormhole nodes that collude with each other, after receiving the RREQ, the wormhole node W_1 will encapsulate the RREQ and deliver it to its partner W_2 via the tunnel between them, and then the wormhole node W_2 can receive the RREQ after the decapsulation. Thus, after the establishment of the routing paths by the routing reply process as shown in **Figure 7**, the routing path with the wormhole nodes W_1 and W_2 is definitely much smaller than the other regular routing paths.

CH does not participate in the routing request process and routing response process, but is defined to only monitor the abnormal occurrence of the cluster. After the routing path is established, each intermediate node reports the values of Hop_{RREQ} and Hop_{RREP} to the corresponding CH, as shown in 8. You will see the cluster monitored by CH_1 and the same process will happen on other clusters as well. Each CH then calculates the hop value ($Hop = Hop_{RREQ} + Hop_{RREP}$) of the routing path to each node in the cluster to find the maximum hop value (Hop_{max}). And the minimum hop value (Hop_{min}). If the gap (G) between two hop count values is greater than a predefined threshold (Hop_{th}), CH determines that there is a wormhole node in the cluster. The size of the threshold Hop_{th} is given based on the density and the scope of the MANET. If the network is dense and large, a relatively small value is given to the threshold Hop_{th}. Otherwise, if the network is sparse and small, assign

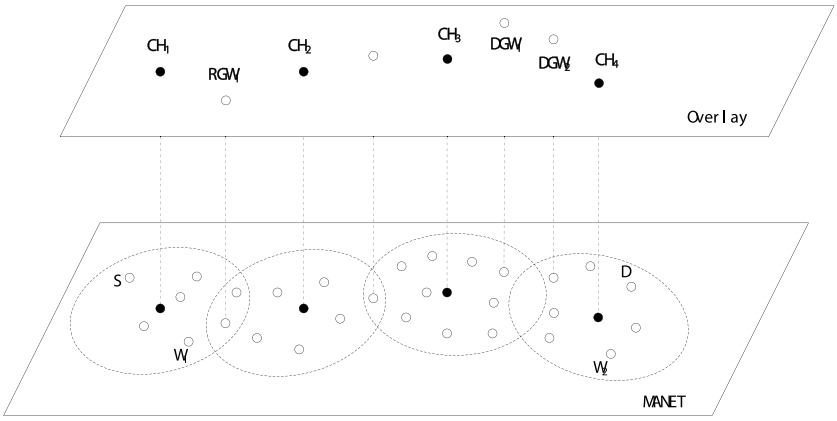

FIGURE 5
Clusters and overlay layer with clusterheads and gateways.

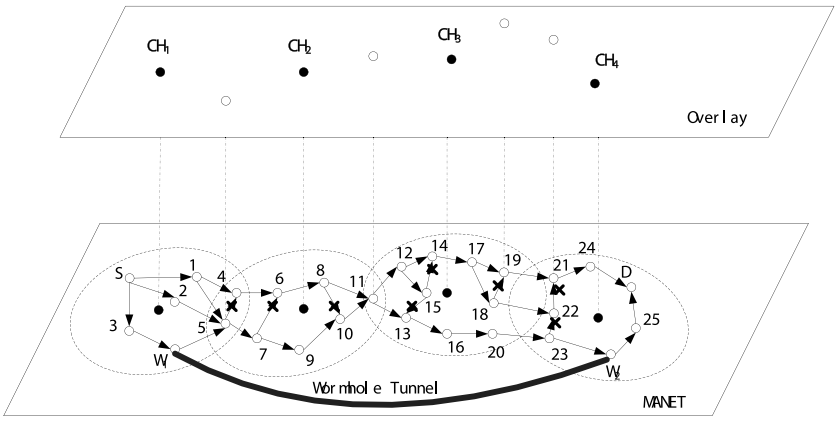

FIGURE 6
Routing request process.

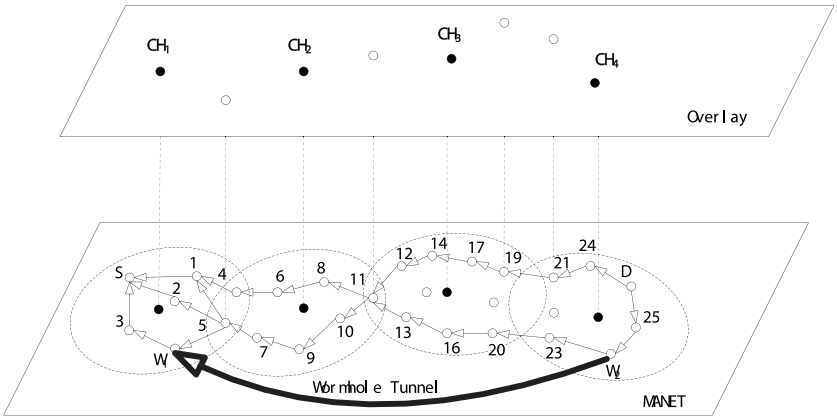

FIGURE 7
Routing reply process.

a relatively large value to the threshold Hop_{th}. The way for computing threshold Hop_{th} is as Eq. 18.

In our scheme, it is stipulated that only the destination node has the right to reply RREP to RREQ. In contrast to AODV, any intermediate node with a path to the target node can reply to RREQ using RREP, which is forbidden in our scheme. In the case of AODV, detection of a wormhole attack is not guaranteed, because wormhole nodes may not participate in routing paths and intermediate nodes that reply to RREPs. In contrast, in our scheme all nodes are required to participate in routing process to expose the existence of wormhole nodes. In routing request process, each node records hop counts from RREQ to the source when it passes through the routing path named as Hop_{RREQ}. On the other hand, in the route reply process, while RREP passes through the reverse routing path, each node records hop counts to the destination, which is named as Hop_{RREP}, like ??

As in Figure 6, RREQ resented by node five is dropped by node 4, because the first node of RREQ is node 1, which is the same as the RREQ received by node four from node 1, thus broadcast storms caused by potential loops between Node 1, Node 4, and Node five are avoided. Since it is assumed that the nodes W_1 and W_2 are wormhole nodes that collude with each other, the wormhole node W_1 will encapsulate the RREQ after receiving the RREQ and pass it to its partner W_2 through the tunnel, and then the wormhole node W_2 can receive the decapsulated RREQ. Therefore, after the routing path is established by the routing reply process shown in Figure 7, the routing paths of the wormhole nodes W_1 and W_2 must be much smaller than other conventional routing paths.

It is legitimate that the CHs do not participate in routing request process and also routing response process, but is only responsible for monitoring the abnormal situation in the corresponding cluster. After the routing path is established, each intermediate node reports its own Hop_{RREQ} and Hop_{RREP} values to its corresponding CH, as shown in Figure 8, in which only the first The details in this show the cluster monitored by CH_1, and the same process happens in other clusters. After that, each CH computes the Hop count value of the routing path in its cluster ($Hop = Hop_{RREQ} + Hop_{RREP}$), and obtains the maximum Hop count value (Hop_{max}) and the minimum hop value (Hop_{min}). If the gap (G) between the two Hop count values is greater than a predefined threshold (Hop_{th}), then CH judges that there are wormhole nodes in its cluster. The size of the threshold Hop_{th} is given based on the density and scale of the MANET. If the network is dense and large, the threshold Hop_{th} is given a relatively low value; otherwise, if the network is sparse and large in scale small, assign the threshold Hop_{th} to a relatively large value. The method to compute the threshold Hop_{th} is in Eq. 18.

$$Hop_{th} = \lceil \frac{D}{R} \rceil \tag{18}$$

$$R = m \cdot \sqrt{\frac{k}{\pi \cdot (n-1)}} \tag{19}$$

where D is the distance between the start and end nodes. R is the average distance of each hop and is calculated by the relationship between density and number of nodes in Eq. 19. m is the length of square side. n is the number of nodes in the network. In our scheme, the threshold Hop_{th} is set to 5, which means that if the gap (G) between Hop_{max} and Hop_{min} is greater than 5, then the CH completes in error. During the vulnerability detection phase, the presence of a wormhole attack is locally detected by the CH, and the possible candidates for one of the wormhole nodes are narrowed down to that cluster member.

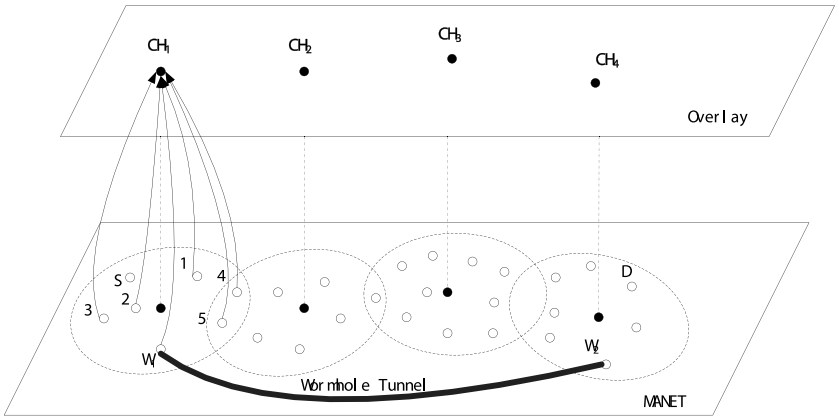

FIGURE 8
Clusterheads monitor the corresponding clusters.

5 Location phase for locating wormhole nodes

After CHs detects that there is a wormhole attack in the corresponding cluster, it triggers the wormhole node localization phase. Each node in each cluster needs to report its one-hop neighbor node list to the corresponding CH. One-hop neighbor information can be obtained through beacons (HELLO messages). By means of a watchdog mechanism, forging neighbor lists can be avoided. Finally, CH can collect a list of neighbor nodes for each node. After that, CH performs the following steps.

- Step 1: If neighbors of a member node are all in the cluster, the CH considers the member node to be a legitimate node. CH continues to check the remaining member nodes until it finds a node with neighbors that are not in the cluster, then CH proceeds to step 2.
- Step 2: After identifying a neighbor node in the neighbor list that is not in the corresponding cluster, the CH sends a topology check (Topology Check, TC) message to the neighbor CH of the neighbor cluster through the GW to track the neighbor node. Indicates that in addition to the responsibility of locally monitoring each cluster, another responsibility of the CH is to verify the local topology through TC messages.
  - If an adjacent node of the member node in the previous cluster is found in the adjacent cluster, the adjacent CH replies to the TC message with a topology check reply (TCR) message to confirm the legitimacy of the member node in the previous cluster.
  - If no neighbor of the member node in the previous cluster exists in the neighbor cluster, the neighbor CH does not reply to the TCR.
- Step 3: After waiting for a predetermined time, if the CH that sent the TC does not receive any TCR, the corresponding member node is regarded as a node forging adjacent nodes, that is, a wormhole node.

For example, as shown in **Figure 9**, node W_1 claims that node W_2 is an adjacent node, satisfying the definition of a wormhole attack. However, the corresponding clusterhead CH_1 cannot find node W_2 in its cluster, so clusterhead CH_1 sends a TC message to its neighboring cluster through the GW. The adjacent CH, namely CH_2, also cannot find the existence of node W_2. Therefore, after waiting for some time, CH_1 cannot receive any TCRs from its neighboring CHs, so CH_1 identifies node W_1 as a wormhole node in its cluster. The positioning process of W_2 is the same as that of the wormhole node W_1. After the location of the wormhole node W_1, CH_1 sends an alert message through the overlay to warn the rest of the network to expose the wormhole node.

The reasons for dividing worm attack prevention into two phases, the detection phase and the localization phase, are as follows. The goal is to apply precautions against worm attacks locally rather than globally (i.e. in each cluster). The discovery phase is performed only on each cluster, ensuring locality. The locality of our plans can significantly reduce overhead and ensure plan scalability, especially in dense and large MANETs. For the localization stage, we separate the localization and detection stages because the localization does not reach 100%. However, since only TC messages and TCR messages are transmitted between adjacent clusters even in the positioning phase, the overhead does not increase much.

6 Simulation results

In this section, the performance is simulated. The network consists of 50 nodes in random.

In **Figure 10**, if it has no wormhole attack in the network, our scheme hardly degrades the network throughput. When a wormhole attack occurs on the network and the traffic load is less than 650 Kbps, the performance of our mechanism is unaffected compared to the one without wormhole attack. However, under the wormhole attack, using the AODV protocol, the delivery rate is only about 52% regardless of the traffic load. This is because about half of the traffic goes through the path of the wormhole link. When using our scheme, wormhole attacks can be detected and wormhole nodes can be located, so the performance is almost the same as without wormhole attacks. The performance of our scheme gradually degrades when the traffic load exceeds 750 Kbps. This is caused by the additional overhead involved by the scheme. It is concluded that our scheme is sufficient to prevent wormhole attacks when the traffic load is moderate, i.e., the maximum traffic load is around 750 Kbps.

The proposed scheme can not only detect wormhole attacks, but locate wormhole nodes. The scheme also considers more parameters into the calculation of node's weight. The purpose of including more parameters is to enhance the cluster node's stability and reduce the frequencies of reelection of the cluster node. This scheme also chooses more important and rightful parameters in order to make the balance between the rightness of the selection of cluster node and efficiency of weight calculation and exchange.

Furthermore, as the time goes by, the performance of our scheme is not decreases. In our scheme, each CH takes charge of each cluster so that the process of attack detection is performed locally, by which the overhead can be reduced. Therefore, the performance of our scheme almost keeps stable.

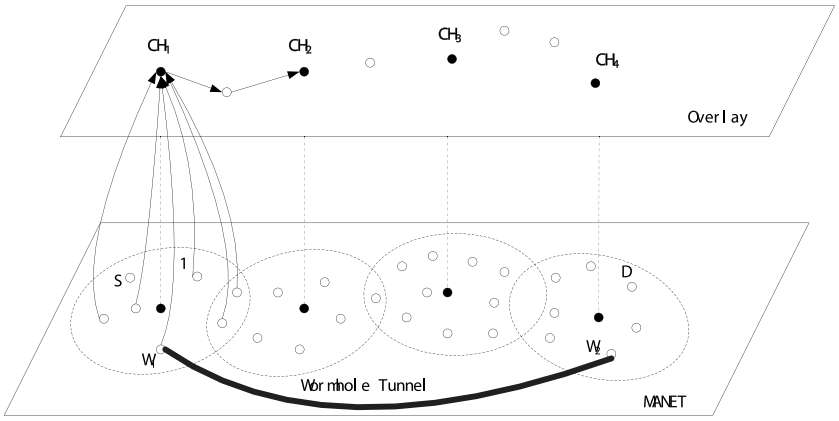

FIGURE 9
Location phase for locating the wormhole nodes.

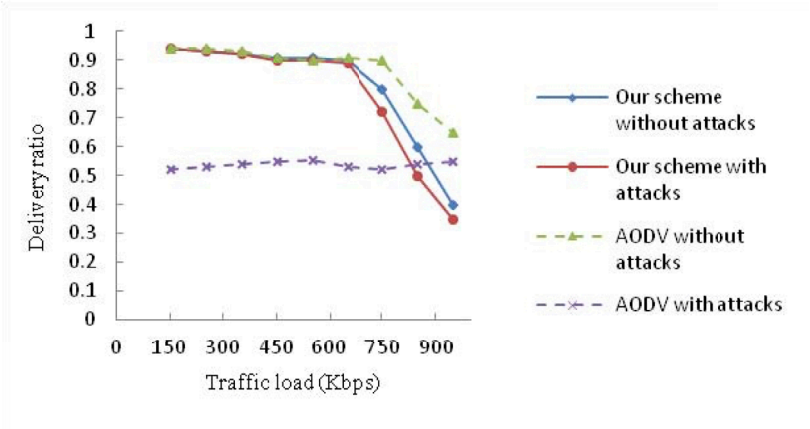

FIGURE 10
Packet delivery ratio vs traffic load.

7 Conclusion

In this paper, a new cluster-based scheme is proposed to combat wormhole attacks in MANET. Performance simulations confirm the usability and efficiency of our scheme. In our future work, we will compare our scheme with more relevant ones. Furthermore, we plan to port the proposed scheme to prevent other types of attacks.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

WL, first author, propose the main scheme in this paper. ZC, XY and XZ are co-authors who do the paper work and simulations.

Funding

This research was supported by National Social Science Fund Project through the research on construction mode of Yangtze River Delta regional innovation and entrepreneurship ecosystem in the digital economy era (21BGL059), and also supported by Key Project of Zhejiang Soft Science Research Program through Research on the Construction

Thought of Zhejiang Digital Technology Innovation Center (2019C25034).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Adarsh, A., Tamang, T. L., Pradhan, P., Singh, V. K., Sen, B., and Sharma, K. (2021). Delay-based approach for prevention of rushing attack in MANETs. *Lect. Notes Netw. Syst.* 281, 125–138. doi:10.1007/978-981-16-4244-9_10

Afzal, Z., and Kumar, M.. Security of vehicular ad-hoc networks (MANET): A survey. *J. Phys. Conf. Ser.* 1427, 0120152020.

Ahutu, O. R., and El-Ocla, H. (2020). *Centralized routing protocol for detecting wormhole attacks in wireless sensor networks*. IEEE Access, 63270–63282.

Garg, S., Singh, A., Kaur, K., Singh Aujla, G., Batra, S., Neeraj Kumar, M. S., et al. (2019). Edge computing-based security framework for big data analytics in VANETs. *IEEE Netw.* 33 (2), 72–81. doi:10.1109/mnet.2019.1800239

Ghugar, U., and Pradhan, J. (2021). Survey of wormhole attack in wireless sensor networks. *Comput. Sci. Inf. Technol.* 2 (1), 33–42. doi:10.11591/csit.v2i1.p33-42

He, W., Lu, M., Zheng, Y., and Xiong, N. N. (2022). *Research on graph structure data adversarial examples based on graph theory metrics*. Cham: Springer.

Hu, X., Hu, Z., Jiang, J., Xue, W., Hu, X., and Xu, X. (2022). Character embedding-based bi-lstm for zircon similarity calculation with clustering. *Earth Sci. Inf.* 15 (3), 1417–1425. doi:10.1007/s12145-022-00847-y

Krundyshev, V., Kalinin, M., and Peter, Z. (2018). Artificial swarm algorithm for MANET protection against routing attacks. IEEE Industrial Cyber-Physical Systems (ICPS), May 2018, 795–800. doi:10.1109/ICPHYS.2018.8390808

Raghav, L. P., Kumar, R. S., Raju, D. K., and Singh, A. R. (2022). Analytic Hierarchy Process (AHP) – swarm intelligence based flexible demand response management of grid-connected microgrid. *Appl. Energy* 306, 118058. doi:10.1016/j.apenergy.2021.118058

Sarhan, S., and Sarhan, S. (2021). Elephant herding optimization ad hoc on-demand multipath distance vector routing protocol for MANET. *IEEE Access* 9 (99), 39489–39499. doi:10.1109/access.2021.3065288

Tahboush, M., and Agoyi, M. (2021). A hybrid wormhole attack detection in mobile ad-hoc network (MANET). *IEEE Access* 9, 11872–11883. doi:10.1109/access.2021.3051491

Tiado, M. I., Noura, I. G., and Hussein, C. (2021). Quality of service evaluation with DSR (dynamic source routing) protocol in the classroom Ad hoc network of the new generation of digital open universities (DOUNG)." in International Conference on Soft Computing and Pattern Recognition. Springer.