A countermeasure against wormhole attacks in MANETs using analytical hierarchy process methodology

Fei Shi · Weijie Liu · Dongxu Jin · Jooseok Song

Published online: 11 May 2013
© The Author(s) 2013. This article is published with open access at Springerlink.com

Abstract Mobile ad hoc networks are vulnerable to a large group of attacks, e.g., wormhole attacks. In this paper, we propose a countermeasure to prevent wormhole attacks. We utilize analytical hierarchy process to elect some special nodes, named the local most trustable nodes, for the source and the destination node, respectively. The elected nodes are then required to implement our proposed scheme to prevent wormhole attacks. The proposed scheme cannot only detect wormhole attacks, but also locate wormhole nodes, i.e., identify the malicious nodes that behave wormhole attacks. To solve the colluding wormhole attack, we present a countermeasure named bi-directional wormhole location mechanism.

Keywords Mobile ad hoc networks · Wormhole · Analytical hierarchy process · Local most trustable node

1 Introduction

A mobile ad hoc network (MANET) [1–5] consists of a collection of wireless mobile nodes that communicate with each other without the use of any network infrastructure. Compared to other types of wireless networks, MANETs are more vulnerable
to a large group of attacks due to their unique characteristics [6]. In this paper, we focus on preventing wormhole attacks. In wormhole attacks, a wormhole consists of two colluding malicious nodes so-called wormhole nodes, which are usually far from each other, and a tunnel between them. The tunnel could be a dedicated communication medium such as long range wireless devices [7]. One wormhole node captures routing traffic at one point of the network and tunnels them to its peer wormhole node at another point. Hence, the network topology is corrupted and routing is compromised. Because wormhole nodes need not to modify or create new packets, no cryptographic technique [8, 9] can prevent wormhole attacks.

A wormhole node ($W_1$) encapsulates packets and sends them to its peer wormhole node ($W_2$) through the path between them. Afterward, $W_2$ obtains the original packets extracted from the encapsulated packets, which is known as decapsulation. Because the original packets are encapsulated, they are not changed by intermediate nodes along the path between $W_1$ and $W_2$. Hence, it seems as if $W_2$ got the packets directly from $W_1$ with the same hop count [10] although they are commonly many hops far away from each other. Hence, the path with wormhole nodes are likely shorter than other normal paths. Therefore, senders prefer to choose the path with wormhole nodes as the routing path than other normal paths to transmit packets.

In this paper, we present a countermeasure against wormhole attacks in MANETs. We list the innovations and contributions of our paper as follows: (1) Special hardware, e.g., directional antenna or synchronized clock, is not required. This merit makes the proposed scheme more suitable for MANETs and easier to be employed in realistic networks. (2) In MANETs, it is unreasonable to treat any node to be secure without any proof. However, in many literatures, either source or destination is assumed to be secure and treated as a monitor or a detector for detecting wormhole attacks. The proposed scheme conforms to the requirement of MANETs, i.e., every node has the uniform security level. In the proposed scheme, we treat source and destination as normal nodes without security priority. (3) The proposed scheme can not only detect wormhole attacks but also locate the wormhole nodes. (4) We originally introduce LMT nodes of source and destination, respectively, so as to implement the prevention of wormhole attacks. By this tactic, the assumption that source or the destination is always secure is not required. At the same time, by using AHP methodology to elect LMT nodes, the security priority of the LMT nodes is reasonable.

The organization of the paper is as follows. Section 2 discusses the related work. Section 3 proposes an algorithm using analytical hierarchy process (AHP) methodology to elect two local most trustable (LMT) nodes for the source and the destination, respectively, which implement the prevention of wormhole attacks. Section 4 presents the details of the countermeasure. Section 5 concludes the paper and presents our future works.

## 2 Related work

In [11], two mechanisms are introduced to detect wormhole attacks: temporal packet leashes and geographical packet leashes. In temporal leashes, accurate clock synchronized clocks are used to restrict the propagation time of packets. In geographic
leashes, loose clock synchronization and location information are used to restrict the migration distance of packets. However the clock synchronization and location information must be obtained via extra hardware, e.g., GPS or other positioning systems [12–14]. Moreover, both temporal and geographical leashes are required to add authentication information to each packet, which take up huge amounts of storage.

In [15], directional antennas are used to prevent wormhole attacks. Nodes use directional antennas to transmit packets to their neighbor nodes in a particular direction. It is also assumed that all antennas on nodes are aligned. The process of neighbor discovery is implemented in a secure way using directional antennas. However, it is probably infeasible to deploy directional and aligned antennas on all of mobile nodes in practice.

In [16], a statistical analysis of multipath (SAM) is proposed to detect wormhole attacks in the network adopting multi-path routing protocol. Due to tunneling by wormhole nodes, the number of hops of the path with wormhole nodes appears to be smaller than normal paths. Thus, the routing path with the wormhole nodes is more attractive to routing discovery of the sources. Through statistics calculation of relative frequency of each routing path, the path that has the biggest relative frequency is identified as the path with the wormhole nodes. However, the drawback is that, in non-multipath routing protocol e.g., AODV, this proposal cannot work.

In [17], the authors propose a secure routing protocol based on the AODV routing protocol: Wormhole Avoidance Routing Protocol (WARP). It does not require any hardware. It considers link-disjoint multi-paths during path discovery, and provides greater path selections to avoid malicious nodes, but eventually uses only one path to transmit data. Based on the characteristic that wormhole nodes can easily grab the route from source node to destination node, WARP enables the neighbors of the wormhole nodes to discover that the wormhole nodes have abnormal path attractions. Then, the wormhole nodes would be gradually isolated by their normal neighboring nodes, and finally be quarantined by whole network. However, some nodes may be misjudged to be wormhole nodes because they are located at the key positions of connectivity within the network.

In our previous work [18], we proposed a time-based mechanism to prevent wormhole attacks in MANETs. Although, special hardware or synchronization system is not required in [18], we made an assumption in [18] which is not always realistic that the source node and the destination node are trustable. However, the assumption is not quite in accord with actual conditions, i.e., the requirement of MAENTs requires that each node in MAENTs has the same security level. In our proposed scheme, one of the main improvements is that we treat the source node and the destination node as normal nodes without any extra assumption.

3 Preliminaries: electing LMT node

A LMT node is the node with the largest weight value in the vicinity of the source or the destination. The decision for the election of LMT node involves many metrics, including relative stability ($S_r$), credit value ($C_v$) and reciprocal of forward rate ($R_f$), which need to be traded off, as shown in Fig. 1. We use AHP methodology, a theory of measurement through pair-wise comparisons and priority scales, to elect LMT node.
3.1 Preparatory work

Firstly, we propose a novel algorithm using AHP methodology [19] to calculate weight values [20] of nodes in the neighborhood of the source and the destination. The nodes with maximum weight values in the vicinity of the source and the destination are elected as the LMT nodes for the source and the destination, respectively. In this subsection, we introduce three parameters which together determine the weight values of the nodes in the vicinity of the source and the destination, including $S_r$, $C_v$ and $R_f$. $S_r$ is evaluated based on the change rate of neighbors. It is a relative value as the name of the parameter implies. It indicates if the node moves relatively fast or slowly or even stable compared with its neighbors. $C_v$ is evaluated based on the packet transmission behavior of nodes. It indicates whether the nodes behave misbehavior by dropping packets or not. $R_f$ is evaluated based on the packet forwarding rate. It indicates whether the nodes violate the backoff mechanism [21] specified by DCF in 802.11 so that they have the higher opportunity to jam the wireless channel, or the nodes perform wormhole attacks that shrink the hop count of the routing path to trap the passing packets. It also indicates the amount of the remaining battery power of node since frequent packet forwarding could enhance the rate of battery power consumption. In brief, the node, which remains relatively stable or moves slowly and exhibits normal behavior and has lower forwarding rate, has the higher opportunity to be elected as the LMT node for the source or the destination. Conversely, the node, which moves fast or performs malicious behavior (i.e., dropping packets) or has larger forwarding rate, has the lower opportunity to be the LMT node.

Due to the limitation of the paper length, in this paper, we do not depict each parameters function and evaluation procedure. One of the calculation methods is proposed in our previous paper [21].

3.2 Calculation of node’s weight

In this subsection, the calculation of weight value of node will be presented by considering the three parameters.
The decision of LMT node election involves many metrics. To do that, they have to be measured, i.e., metrics whose measurements must also be evaluated as to how well they serve the objectives of the decision of LMT node maker. The Analytic Hierarchy Process (AHP) is a theory of measurement through pair-wise comparisons and relies on the judgments of experts to derive priority scales. It is these scales that measure intangibles in relative terms. The comparisons are made using a scale of absolute judgments that represents, how much more one element dominates another with respect to a given attribute. The judgments may be inconsistent, and how to measure inconsistency and improve the judgments, when possible to obtain better consistency is a concern of the AHP. The derived priority scales are synthesized by multiplying them by the priority of their parent nodes and adding for all such nodes.

To make a decision of LMT node election in an organized way to generate priorities, we decompose the decision into the following four steps [19]:

Step 1: Define the problem.

The decision problem is to select a proper node by considering corresponding weight value to be the LMT node among the one-hop vicinity of either source or destination in the environment of MANETs.

Step 2: Structure the decision hierarchy from the top with the goal of the decision, then the objectives from a broad perspective, through the intermediate levels (criteria on which subsequent elements depend) to the lowest level (which usually is a set of the alternatives).

As Fig. 1 shown, the problem is structured as a hierarchy. The overall objective of choosing an appropriate LMT node is placed at the topmost goal of the hierarchy. The subsequent level representing the main criteria are termed as secondary goal. The three secondary goals are $S_r$, $C_v$, and $R_f$. Finally, alternatives are placed in the bottom level, which are evaluated for the selection of LMT node.

Step 3: Construct a set of pair-wise comparison matrices.

To make comparisons, we need a scale of numbers that indicates how many times more important or dominant one element is over another element with respect to the criterion or property with respect to which they are compared. Table 1 exhibits the scale.

Table 1  The fundamental scale of absolute numbers

| Intensity of importance | Definition          | Explanation                                                   |
|-------------------------|---------------------|---------------------------------------------------------------|
| 1                       | Equal importance    | Two activities contribute equally to the objective            |
| 2                       | Weak or slight      | Experience and judgment slightly favor one activity over another |
| 3                       | Moderate importance | Experience and judgment strongly favor one activity over another |
| 4                       | Moderate plus       | An activity is favored very strongly over another; its dominance demonstrated in practice |
| 5                       | Strong importance   | The evidence favoring one activity over another is of the highest possible order of affirmation |

As Fig. 1 shown, the problem is structured as a hierarchy. The overall objective of choosing an appropriate LMT node is placed at the topmost goal of the hierarchy. The subsequent level representing the main criteria are termed as secondary goal. The three secondary goals are $S_r$, $C_v$, and $R_f$. Finally, alternatives are placed in the bottom level, which are evaluated for the selection of LMT node.

Step 3: Construct a set of pair-wise comparison matrices.

To make comparisons, we need a scale of numbers that indicates how many times more important or dominant one element is over another element with respect to the criterion or property with respect to which they are compared. Table 1 exhibits the scale.

Table 1  The fundamental scale of absolute numbers

| Intensity of importance | Definition          | Explanation                                                   |
|-------------------------|---------------------|---------------------------------------------------------------|
| 1                       | Equal importance    | Two activities contribute equally to the objective            |
| 2                       | Weak or slight      | Experience and judgment slightly favor one activity over another |
| 3                       | Moderate importance | Experience and judgment strongly favor one activity over another |
| 4                       | Moderate plus       | An activity is favored very strongly over another; its dominance demonstrated in practice |
| 5                       | Strong importance   | The evidence favoring one activity over another is of the highest possible order of affirmation |
The reciprocal matrix is constructed through pair-wise comparison of each criterion against another that exists under the topmost goal. The values of the pair-wise comparison matrices are provided through answering the questions which has got more preference and, by how much. The following criteria matrix $A$ gives the pair-wise comparison results of three criteria towards the top most goals. The value $a_{ij}$ denotes the strength of preference of $i$th criteria over $j$th criteria. The fundamental 1 to 9 scale is utilized to express the strength of preference based on the intuition, experience and knowledge and it is shown in Table 1. The $A$ matrix is as follows in Eq. (1):

$$
A = [a_{ij}] = \begin{bmatrix}
S_r \\
C_v \\
R_f
\end{bmatrix} = \begin{bmatrix}
1 & a_{S_rC_v} & a_{S_rR_f} \\
1/a_{S_rC_v} & 1 & a_{C_vR_f} \\
1/a_{S_rR_f} & 1/a_{C_vR_f} & 1
\end{bmatrix}
$$

By using mean normalization of row vector, the $A$ can be standardized to a normalized vector matrix $A_{\text{norm}}$ as shown in Eq. (2).

$$
A_{\text{norm}} = \left[ \frac{a_{ij}}{\sum_{i=1}^{k} a_{ij}} \right]
$$

where $k$ is the number of criteria. In the proposed scheme, three parameters are referred as the criteria for evaluating the weight value of node, so $k$ is equal to 3.

Then, by the mean normalization of row vector, we can get the normalized vector $W_i^T$, as shown in Eq. (3), which stands for the weight factoring of each criterion.

$$
W_i^T = [w_j] = \left[ \frac{1}{k} \sum_{j=1}^{k} (a_{ij}^{\text{norm}}) \right]
$$

where $a_{ij}^{\text{norm}}$ is the $i$th row and $j$th column element of $A_{\text{norm}}$.

All the pair-wise comparison matrices are checked for its consistency. Due to people’s random judgments matrices may be prone to judgments errors that can be detected by Consistency Ratio ($CR$) that is defined as the ratio of Consistency Index ($CI$) to Random Index ($RI$). CI can be calculated by Eq. (4) and it is shown for a criteria matrix $C$ as an instance. After the calculation of weight of each criterion, the consistency should be considered.

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

where $n$ denotes the number of elements to be compared in criteria matrix $A$, here it is 3. $\lambda$ can be calculated by the following Eq. (5).

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

where $\mu_i$ is the consistency vector which can be calculated by the following Eq. (6).

$$
\mu_i = \frac{\sum_{j=1}^{n} W_j a_{ij}}{W_i}
$$
A countermeasure against wormhole attacks in MANETs

Table 2  Random index

| Exponent number | 1 | 2 | 3 | 4 | 5 | 6 |
|-----------------|---|---|---|---|---|---|
| RI              | 0 | 0 | 0.58 | 0.90 | 1.12 | 1.24 |

where \( w_i \) is the weight factoring of each criterion calculated by the aforementioned Eq. (3).

Finally, we can obtain the consistency ratio (\( CR \)) which is the ratio between \( CI \) and \( RI \), as shown in Eq. (7). \( RI \) is shown in Table 2.

When \( CR < 0.1 \), the consistency of matrix is really high and acceptable, i.e., judgment errors are tolerable, otherwise, the pair-wise matrix undergoes some adjustments until the matrix satisfies the consistency check.

\[
CR = \frac{CI}{RI} \tag{7}
\]

Step 4: For each criterion in the second level, we need to compute weights for nodes under respective criterion by using same way applied in step 3. Use the priorities obtained from the comparisons to weigh the priorities in the level immediately below. Do this for every element. Then for each element in the level below add its weighed values and obtain its overall or global priority. Continue this process of weighing and adding until the final priorities of the alternatives in the bottom most level are obtained. The local weight is computed for each criterion and alternatives.

After calculating the weight of each criterion, we should calculate the weight of node in the same calculation process. In our LMT node election process, three criteria are considered, i.e., \( S_r \), \( C_v \) and \( R_f \). We need compare nodes in the vicinity from every perspective of each criterion. Hence, we can obtain the following three matrix, i.e., \( A_{S_r} \), \( A_{C_v} \) and \( A_{R_f} \), which are local weights factor of the corresponding parameters, as shown in Eqs. (8), (9), and (10). To make a clear and intuitive explanation, we assume there are only three nodes in the vicinity, i.e., node 1 \( (n_1) \), node 2 \( (n_2) \), and node 3 \( (n_3) \). The \( A_{S_r} \) stands for the pairwise comparisons of nodes in vicinity according to \( S_r \). The \( A_{C_v} \) stands for the pairwise comparisons of nodes in vicinity according to \( C_v \). The \( A_{R_f} \) stands for the pair-wise comparisons of nodes in vicinity according to \( R_f \).

\[
A_{S_r} = \left[ a_{ij}^{S_r} \right] = \begin{bmatrix}
S_{n_1n_1} & S_{n_1n_2} & S_{n_1n_3} \\
1/a_{n_1n_2} & a_{n_2n_2} & a_{n_2n_3} \\
1/a_{n_1n_3} & 1/a_{n_2n_3} & a_{n_3n_3}
\end{bmatrix} \tag{8}
\]

\[
A_{C_v} = \left[ a_{ij}^{C_v} \right] = \begin{bmatrix}
C_{n_1n_1} & C_{n_1n_2} & C_{n_1n_3} \\
1/a_{n_1n_2} & C_{n_2n_2} & a_{n_2n_3} \\
1/a_{n_1n_3} & 1/a_{n_2n_3} & C_{n_3n_3}
\end{bmatrix} \tag{9}
\]

\[
A_{R_f} = \left[ a_{ij}^{R_f} \right] = \begin{bmatrix}
R_{n_1n_1} & R_{n_1n_2} & R_{n_1n_3} \\
1/a_{n_1n_2} & R_{n_2n_2} & a_{n_2n_3} \\
1/a_{n_1n_3} & 1/a_{n_2n_3} & R_{n_3n_3}
\end{bmatrix} \tag{10}
\]
By using the similar process of calculation of $W^T_i$, we can obtain every node’s weight factor on the corresponding criterion as follows in Eq. (11):

$$
\alpha = [\alpha_{ij}] = 
\begin{bmatrix}
\alpha_{n1}^{S_r} & \alpha_{n2}^{S_r} & \alpha_{n3}^{S_r} \\
\alpha_{n1}^C & \alpha_{n2}^C & \alpha_{n3}^C \\
\alpha_{n1}^{R_f} & \alpha_{n2}^{R_f} & \alpha_{n3}^{R_f}
\end{bmatrix}
$$

where the $\alpha_{n1}^{S_r}$ stands for if only $S_r$ is considered.

$$
W_i = \sum_{j=1}^{n} w_j \cdot \alpha_{ij}
$$

The global weight of a mobile node is achieved through multiplying its local weight by its corresponding parent weights. Hence, finally by Eq. (12), we can obtain the global weight vector as follows in Eq. (13):

$$
W_i = \begin{bmatrix} W_{n1} & W_{n2} & W_{n3} \end{bmatrix}
$$

where $W_{n1}$ is the weight of node 1 by considering all of the criteria. For example, $W_{n1} = W_{S_r} \cdot \alpha_{n1}^{S_r} + W_{C_r} \cdot \alpha_{n1}^C + W_{R_f} \cdot \alpha_{n1}^{R_f}$.

Since we’ve already obtain the global weight of each node in the vicinity of the source and the destination, the LMT nodes can be elected. For instance, $n_3$ has the largest weight value, i.e., $W_{n3}$ is the largest one among $W_{n1}$, $W_{n2}$ and $W_{n3}$; therefore $n_3$ is elected as the LMT node.

### 4 Countermeasures against wormhole attacks

By the aforementioned election algorithm, the node in the vicinity of the source or the destination with the largest weight value is elected as the corresponding LMT node. The LMT nodes then perform the countermeasure to prevent wormhole attacks. In this section, we present the countermeasure consisting of three phases. The first one is detection phase that detects the existence of wormhole attacks. If wormhole attacks are detected, the second phase, i.e., location phase described in Sect. 4.2, is triggered to identify the specific wormhole nodes. The third phase, i.e., bidirectional location described in Sect. 4.3, is presented to solve the collaborative wormhole issue.

#### 4.1 Phase one: detecting wormhole attacks

As mentioned in Sect. 1, wormhole nodes use tunnel to connect each other to make the routers between two wormhole nodes invisible. Thus, the hop count of the path with wormhole attacks usually appears to be smaller than other normal paths. Consequently, the paths with wormhole attacks are more attractive to the source. Although we assume the basic routing protocol employed in the network is AODV [22], the proposed scheme can be extended to other routing protocols, e.g., DSR, OLSR. The
A countermeasure against wormhole attacks in MANETs

The proposed scheme detects wormhole attacks by comparing two types of hop count. One is extracted from the head of Route Reply (RREP) packet which is polluted by wormhole nodes. In AODV, RREP is sent by the destination after receiving RREQ (Route Request) packet sent by the source. Another hop count is estimated by us, which is the approximation of the actual number of hops between the source and the destination.

**Step 1: Timer setting.**

After the routing discovery process by AODV between the source and the destination, the routing path is selected. Before data transmission through the selected routing path, the LMT node in the vicinity of the source sends a HELLO message whose destination is the LMT node of destination node via the selected routing path. The LMT node of the source sets a timer when it sends the HELLO message. Once the LMT node of the destination receives the HELLO, it is required to send a HELLO REPLY message back to the LMT node of the source via the inverse routing path. When the LMT node of the source receives the HELLO REPLY, it stops the timer. The period from sending the HELLO to receiving the HELLO REPLY is the round trip time (RTT) between the source and destination, donated as $T_{RTT}$.

**Step 2: Minimum hop estimating.**

In this step, the minimum hop count between the source and the destination is estimated.

First, the distance ($D$) between the source and the destination can be evaluated by Eq. (14).

$$D = \frac{(T_{RTT} - 4 \times \frac{\bar{R}}{v}) \times v}{2} + \delta = \frac{T_{RTT} \times v - 2 \times \bar{R}}{2} + \delta$$

where $\delta$ is the maximum relative error caused by the channel condition, and $v$ is the speed of light. $\bar{R}$ is the mean value of node’s transmission range.

Then, the estimated hop count ($H$) between the source and the destination can be calculated by Eq. (15).

$$H \geq \left\lceil \frac{D}{\bar{R}} \right\rceil \geq \left\lceil \frac{T_{RTT} \times v - 4 \times \bar{R}}{2\bar{R}} \right\rceil \geq \left\lceil \frac{T_{RTT} \times v}{2r} - 2 \right\rceil$$

where $r$ is the upper bound of transmission range.

Afterwards, the estimated lower bound of hop count ($H_{min}$) can be presented by Eq. (16).

$$H_{min} = \left\lceil \frac{T_{RTT} \times v}{2r} - 2 \right\rceil$$

**Step 3: Hop testing.**

$H_r$ denotes the hop count extracted from the received RREP. If there are wormhole attacks in the selected routing path, $H_r$ cannot represent the actual hop count between the source and the destination, since wormhole tunnel makes the nodes between the source and the destination invisible. Once the LMT node of source estimates $H_{min}$, it compares $H_r$ with $H_{min}$, if $H_r < H_{min}$, the LMT node of the source can judge
that there is wormhole attacks in the selected routing path; otherwise, the originator selects a shortest path among those routes with the property that $H_r \geq H_{\text{min}}$.

4.2 Phase two: locating wormhole nodes

Once wormhole attack is detected by the LMT node of the source in phase one, the location phase is triggered. The purpose of location phase is to identify wormhole nodes.

Step 1: Timer setting.

The LMT node of the source temporarily enables the path with wormhole attacks and sends a TRACE message along the selected routing path towards the LMT node of the destination. Meanwhile, a timer is initialized by the LMT node of the source. For each intermediate node in the route path, when it receives the TRACE, it replies a TRACE REPLY message to the LMT node of the source via the inverse path piggybacking its hop count. Meanwhile, the TRACE is forwarded by each intermediate node via the route path. When the LMT node of the source receives the TRACE REPLY from each intermediate node, the RTT between the source and each intermediate node can be evaluated by referring to the timer. A scenario is given in Fig. 2 as an example. For each intermediate node ($A, W_1, W_2, B$), after receiving TRACE, each of them reply a TRACE REPLY piggybacking the corresponding hop count. Note that this piggybacked hop count could be polluted by wormhole nodes using the tunneling.

Step 2: Hop testing.

After receiving each TRACE REPLY, the LMT node of the source estimates the corresponding increase of hop count for each intermediate node. If the increase of hop count at one intermediate node is not the one comparing to its previous hop, the current node and its previous node are identified as the wormhole nodes.

To illustrate this process intuitively, we use the scenario in Fig. 2. The LMT node of source $S$ sends a TRACE. The intermediate node $A$ then receives it and replies a TRACE REPLY to $S$ piggybacking $A$’s hop count ($H_A$). Meanwhile, $A$ relays TRACE to its next-hop neighbor towards destination. As to other intermediate nodes ($W_1, W_2$ and $B$), the same process is continued until TRACE reaches the LMT node of destination.

![Fig. 2 Wormhole location mechanism](image)
We take \( W_2 \) as an example. The distance \( (D_{sw2}) \) between source \( S \) and intermediate node \( W_2 \) can be estimated by the LMT node of the source using Eq. (17) that is similar with Eq. (14).

\[
D_{sw2} = \frac{(T_{SW_2} - 2 \times \frac{r}{v}) \times v}{2} + \delta = \frac{T_{SW_2} \times v - 2 \times r}{2} + \delta
\] (17)

where \( T_{SW_2} \) is the RTT between \( S \) and \( W_2 \).

The minimum hop count \( (H_{min_{sw2}}) \) between \( S \) and \( W_2 \) can be evaluated by Eq. (18) that is similar with Eq. (16).

\[
H_{min_{sw2}} = \left\lceil \frac{T_{SW_2} \times v}{2r} - 1 \right\rceil
\] (18)

The hop count of \( W_2 \) piggybacked by the TRACE REPLY sent by \( W_2 \) is presented as \( H_{W_2} \). If \( H_{W_2} < H_{min_{sw2}} \), the LMT node of source \( S \) can judge that \( W_2 \) and its previous node, i.e., \( W_1 \), are wormhole nodes.

The reason to separate the proposed scheme into detection phase and location phase is that location phase is more time consuming than detection phase. Majority of nodes in the network is normal node and minority is malicious node. Even though the wormhole nodes attract routing traffic, only a few paths are the ones with wormhole nodes. Thus, only when the selected routing path is detected as the path with wormhole nodes, will the location phase be triggered to identify the corresponding wormhole nodes. By this way, the consumption of time and other resources could be reduced, while the wormhole attacks could also be well prevented.

There is a potential issue in the aforementioned location mechanism, which is collaborative wormhole issue. As we mentioned that the intermediate nodes are required to reply TRACE REPLY back to the source node after receiving the TRACE message, so that the source can evaluate the RTT between the source and each intermediate node. However, for the purpose of concealing the behavior of wormhole attacks and avoiding being identified, wormhole nodes may pretend to be normal nodes. Here we present each node’s TRACE REPLY by using “REP” plus “_” plus the identifier of the corresponding node, e.g., \( REP_{\_W_2} \) denotes the TRACE REPLY of \( W_2 \). For example, in Fig. 3, the wormhole node \( W_1 \) could fabricate the \( REP_{\_W_2} \) substituting for \( W_2 \) and send this fake \( REP_{\_W_2} \) to the LMT node of the source at a proper time point. We define proper time point as the time that makes \( T_{SW_2} \) smaller than it should actually be in order to make \( H_{W_2} > H_{min_{sw2}} \). Meanwhile, \( W_2 \) performs in a collaborative manner.

![Fig. 3 Wormhole location mechanism preventing collaborative wormhole](image)
way, i.e., it doesn’t send any TRACE REPLY for the TRACE, but simply forwards the TRACE to its next hop node. Through the fabrication of REP_W2 by node W1, it seems as if W2 were in the vicinity of W1, i.e., W2', as shown in Fig. 3. In addition, by the collaboration between W1 and W2, they could successfully avoid being identified by the LMT node of source using the aforementioned location mechanism in phase two. To solve this issue, we propose the phase three.

4.3 Phase three: bidirectional location mechanism

To solve the collaborative wormhole issue, we proposed a mechanism named bidirectional location as follows:

Step 1, Step 2: Step 1 and step 2 are the same with the corresponding steps (Timer Setting and Hop Testing) in Sect. 4.2. If wormhole nodes can’t be identified, the process goes to Step 3.

Step 3: The LMT node of the source stores intermediate nodes’ RTT information.

Step 4: The LMT node of source sends the intermediates’ RTT information to the LMT node of destination via a secure path. We define secure path as the path without wormholes. There are two cases as follows:

Case 1: The LMT node of the source has multiple paths to the LMT node of the destination besides the path including wormhole nodes, which are built during the routing discovery process. In this case, the LMT node of the source chooses another path without wormhole nodes to transmit the intermediates’ RTT information to the LMT node of the destination.

Case 2: The LMT node of the source node has only one path to the LMT node of the destination, which includes wormhole nodes. In this case, in order to transmit the intermediates’ RTT information to the LMT node of the destination securely, the LMT node of the source is required to initiate a new routing discovery process to find a new path without wormholes and use the path to transmit the intermediates’ RTT information.

Step 5: This step is similar with step 1 in Sect. 4.2 whose function is to evaluate the RTT between the destination and each intermediate node. One difference is that here TRACE is sent by the LMT node of the destination in the inverse direction of the same route path mentioned in both phase one and two. Another difference is that the timers are set by the LMT node of the destination. The other difference is that intermediate nodes reply TRACE REPLY messages to the LMT node of the destination, as shown in Fig. 4.

![Fig. 4 Bidirectional wormhole location](image-url)
Step 6: The LMT node of the destination makes the judgment for identifying wormhole nodes based on the following two sub-steps:

Sub-step 6.1: This sub-step is similar with step 2 in Sect. 4.2 whose function is to specify the wormhole node according to Eqs. (31) and (32). The difference is that the LMT node of the destination performs the judgment. Thus, $T_{SW_2}$ in Eqs. (31) and (32) is replaced by $T_{DW_1}$ that represents the RTT between $D$ and $W_1$; $D_{SW_2}$ in Eq. (31) is replaced by $D_{DW_1}$ that represents the distance between $D$ and $W_1$; $H_{min\_sw_2}$ is replaced by $H_{min\_DW_1}$ that represents the minimum hop count between $D$ and $W_1$. If $H_{W_1} < H_{min\_DW_1}$, $W_1$ and its previous node, i.e., $W_2$, are wormhole nodes; otherwise, the process goes to sub-step 6.2. Herein, $H_{W_1}$ is the hop count between $W_1$ and destination $D$ which is piggybacked in the TRACE REPLY (REP$_{W_1}$) required to be sent by $W_1$ and probably fabricated by $W_2$ to reply the TRACE sent by the LMT node of $D$. Through the fabrication of REP$_{W_1}$ by $W_2$, it seems as if $W_1$ is in the vicinity of node $W_2$, i.e., $W'_1$ in Fig. 4. Hence, the following sub-step 6.2 is required to check the symmetry of the routing path between the source and the destination.

Sub-step 6.2: If wormholes could not be located in sub-step 6.1, it means that the collaboration wormhole attack exists as we explained above in which $W_2$ has fabricated the TRACE REPLY (REP$_{W_1}$) for $W_1$. If $T_{SW_1} + T_{DW'_1} = T_{SD} + \delta$, which means that the path is symmetrical, step 6 is iterated to check the following left nodes until the wormhole nodes are identified; otherwise, if $T_{SW_1} + T_{DW'_1} \neq T_{SD} + \delta$, which means the path is unsymmetrical caused by the fake TRACE REPLY (REP$_{W_1}$) fabricated by the collaborative wormhole nodes, the LMT nodes can identify that $W_1$ and $W_2$ are wormhole nodes. Herein, $T_{SW_1}$ is the RTT between $S$ and $W_1$. $T_{DW'_1}$ is the RTT between $D$ and $W_1$ which is fabricated by the collaborative node $W_2$. $T_{SD}$ is the RTT between $S$ and $D$. $\delta$ is the error of the symmetric path.

Figure 5 concludes the process of the proposed bidirectional wormhole location mechanism.

5 Conclusion and future work

In this paper, we propose a scheme to prevent wormhole attacks in MANETs. The proposed scheme can not only detect wormhole attacks, but locate wormhole nodes. To solve the potential issue in the location phase, i.e., collaborative wormhole, we propose a bidirectional location mechanism.

However, our scheme has an overhead to calculate the weight value. (1) For our LMT election algorithm, when no transmission exists at initial time, the number of packets actually forwarded and the number of packets expected to be forwarded cannot be determined. Thus, the credit value and the reciprocal of forward rate cannot be evaluated at initial time. The values can be evaluated only after a period of transmission, which causes delay in the evaluation of the weight value of each node. (2) When a node moves to a new neighborhood, the node becomes a stranger to its neighbors. Hence, the node’s weight value must be recalculated, and the problem mentioned above may arise again. If the node moves frequently, this problem may become more serious. However, the stability value is a variable of the weight
value function in our scheme. If the node moves frequently, its stability value will be small. Accordingly, the weight value of the node will be small, which means that the node will not become the LMT node with the largest weight value in the neighborhood.

In the near future, our research work will concern about the following aspects: (1) We will consider more parameters into the calculation of node’s weight, like cumulative time during which the node has worked as the LMT node. Cumulative time implies node stability that increases the stability of the LMT node. The purpose of including more parameters is to enhance the LMT node’s stability and reduce the frequencies of reelection of the LMT node. We will measure more parameters and
choose more important and rightful ones in order to make the balance between the rightness of the selection of LMT node and efficiency of weight calculation and exchange. (2) We will apply the proposed scheme on preventing other misbehaviors in MANETs, e.g., blackhole attacks [23] where the malicious node could attract all or most packets and then misuses those packets, and MAC layer attacks [21] where the malicious node could violate the MAC layer protocol. (3) In the near future, we plan to use Network Simulator version 2 (NS-2) to evaluate the performance of the proposed scheme. We will compare it with our previous scheme [18] and AODV [22]. The simulation will be implemented in two scenarios, i.e., network without wormhole attacks and network with wormhole attacks, respectively. In the former scenario, we intend to evaluate how much the network throughput is reduced along with the raise of traffic loads by using our scheme. In the latter scenario, we’d like to measure how much the packets delivery rate is enhanced due to the wormhole prevention by using our scheme.

**Acknowledgements** This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1B3004161).

**Open Access** This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

**References**

1. Isaac, J. T., Zeadally, S., & Cámara, J. S. (2010). Implementation and performance evaluation of a payment protocol for vehicular ad hoc networks. *Journal of Electronic Commerce Research, 10*(2), 209–233.
2. Zhou, X., Ge, Y., Chen, X., Jing, Y., & Sun, W. (2012). A distributed cache based reliable service execution and recovery approach in MANETs. *Journal of Convergence, 3*(1), 5–12.
3. Shbat, M. S., & Tuzlukov, V. (2011). Dynamic frequency reuse factor choosing method for self-organizing LTE networks. *Journal of Convergence, 2*(2), 13–18.
4. Tseng, H., Chou, L.-D., & Chao, H.-C. (2011). A survey of black hole attacks in wireless mobile ad hoc networks. *Human-Centric Computing and Information Sciences, 1*(4). doi: 10.1186/2192-1962-1-4.
5. Isaac, J. T., Zeadally, S., & Cámara, J. S. (2012). A lightweight secure mobile payment protocol for vehicular ad-hoc networks (VANETs). *Journal of Electronic Commerce Research, 12*(1), 97–123.
6. Cho, J. H., Swami, A., & Chen, I. R. (2011). A survey on trust management for mobile ad hoc networks. *Communication Surveys & Tutorials, 13*(4), 562–583.
7. Nagrath, P., & Gupta, B. (2011). Wormhole attacks in wireless ad hoc networks and their counter measurements: a survey. In *3rd international conference on electronics computer technology* (pp. 245–250).
8. Tan, Z. (2012). An efficient identity-based tripartite authenticated key agreement protocol. *Journal of Electronic Commerce Research, 12*(4), 505–518.
9. Misra, S. K., & Wickamasinghe, N. (2004). Security of a mobile transaction: a trust model. *Journal of Electronic Commerce Research, 4*(4), 359–372.
10. Loukola, M. V., & Skyttä, J. O. (2001). Enhanced augmented IP routing protocol (EAIRP) in IPv6 environment. *Journal of Electronic Commerce Research, 1*(4), 359–370.
11. Hu, Y. C., Perrig, A., & Johnson, D. B. (2003). Packet leashes: a defense against wormhole attacks in wireless networks. In *22nd INFOCOM* (pp. 1976–1986).
12. Petrova, K., & Wang, B. (2011). Location-based services deployment and demand: a roadmap model. *Journal of Electronic Commerce Research, 11*(1), 5–29.
13. Aloudat, A., & Michael, K. (2011). Toward the regulation of ubiquitous mobile government: a case study on location-based emergency services in Australia. *Journal of Electronic Commerce Research, 12*(1), 31–74.

14. Zhou, T. (2013). An empirical examination of user adoption of location-based services. *Journal of Electronic Commerce Research, 13*(1), 25–39.

15. Hu, L., & Evans, D. (2004). Using directional antennas to prevent wormhole attacks. In *Network and distributed system security symposium* (pp. 131–141).

16. Qian, L., Song, N., & Li, X. (2005). Detecting and locating wormhole attacks in wireless ad hoc networks through statistical analysis of multi-path. In *IEEE wireless communications and networking conference* (pp. 2106–2111).

17. Su, M.-Y. (2009). WARP: a wormhole-avoidance routing protocol by anomaly detection in mobile ad hoc networks. *Journal of Computers & Security, 29*(2), 208–224.

18. Shi, F., Jin, D., Liu, W., & Song, J.-S. (2011). Time-based detection and location of wormhole attacks in wireless ad hoc networks. In *International joint conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11* (pp. 1721–1726).

19. Saaty, T. L. (2008). Decision making with the analytic hierarchy process. *International Journal of Services Sciences, 1*(1), 83–98.

20. Aikebaier, A., Enokido, T., & Takizawa, M. (2011). Trustworthy group making algorithm in distributed systems. *Human-Centric Computing and Information Sciences, 1*(6). doi: 10.1186/2192-1962-1-6.

21. Shi, F., Baek, J., Song, J., & Liu, W. (2011). A novel scheme to prevent MAC layer misbehavior. *IEEE 802.11 Ad Hoc Networks, Telecommunication Systems*. doi:10.1007/s11235-011-9552-y.

22. Perkins, C. E., & Royer, E. M. (1999). Ad-hoc on-demand distance vector routing. In *2nd IEEE workshop on mobile computing systems and applications* (pp. 90–100).

23. Nadeem, A., & Howarth, M. (2011). Protection of MANETs from a range of attacks using an intrusion detection and prevention system. *Telecommunications Systems*. doi:10.1007/s11235-011-9484-6.

Fei Shi received the B.S. degree in computer science and M.S. degree in computer system from the Northeastern University, Shenyang, China, in 2004, and 2008, respectively. He is currently working toward the Ph.D. degree in computer science at Yonsei University, Seoul, Korea. His research interests include wireless communications, mobile ad hoc networks, vehicular ad hoc networks and information security.

Weijie Liu received the B.S. degree in computer science and M.S. degree in history of science & technology from the China University of Geosciences, Wuhan, China, in 2006, and 2008, respectively. She is currently working toward the Ph.D. degree in the department of information & industrial engineering at Yonsei University, Seoul, Korea. Her research interests include network security, semantic network and semantic technology in wireless communication, artificial intelligence in wireless communication.
Dongxu Jin received the Bachelor’s degree in computer science and technology from the Northeastern University, Shenyang, China, in 2007. And in 2010, he received the M.S. degree in computer science from the Korea University, Seoul, Korea. He is currently a Ph.D. candidate at department of computer science in Yonsei University, Seoul, Korea. His research area includes wireless sensor networks, RFID, wireless communications, mobile ad hoc networks, vehicular ad hoc networks and vertical handover.

Jooseok Song received the B.S. degree in Electrical Engineering from Seoul National University, Seoul, Korea, in 1976, and the M.S. degree in Electrical Engineering from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1979. In 1988, he received the Ph.D. degree in Computer Science from University of California at Berkeley. From 1988 to 1989, he was an Assistant Professor at the Naval Postgraduate School, Monterey, CA. He was the president of Korea Institute of Information Security and Cryptology in 2006. He is currently a Professor of computer science at Yonsei University, Seoul. His research interests include cryptography and network security.