TOHIP: A topology-hiding multipath routing protocol in mobile ad hoc networks

Yujun Zhang\textsuperscript{a}, Tan Yan\textsuperscript{b,*}, Jie Tian\textsuperscript{b}, Qi Hu\textsuperscript{a}, Guiling Wang\textsuperscript{b}, Zhongcheng Li\textsuperscript{a}

\textsuperscript{a} Institute of Computing Technology, Chinese Academy of Sciences, Beijing, PR China
\textsuperscript{b} Department of Computer Science, New Jersey Institute of Technology, Newark, USA

\begin{abstract}
Existing multipath routing protocols in MANETs ignore the topology-exposure problem. This paper analyzes the threats of topology-exposure and proposes a TOpology-HIding multipath Protocol (TOHIP). TOHIP does not allow packets to carry routing information, so the malicious nodes cannot deduce network topology and launch various attacks based on that. The protocol can also establish multiple node-disjoint routes in a route discovery attempt and exclude unreliable routes before transmitting packets. We formally prove that TOHIP is loop-free and does not expose network topology. Security analysis shows that TOHIP can resist various kinds of attacks efficiently and effectively. Simulation results demonstrate that TOHIP has better capability of finding routes and can greatly increase the capability of delivering packets in the scenarios where there are malicious nodes at the cost of low routing overhead.
\end{abstract}

1. Introduction

Multipath routing protocols have attracted a lot of attentions recently in MANET for their unique capability in supporting load balancing and improving routing reliability in high dynamic scenarios \cite{1,2}. However, this kind of protocol may become a vulnerable target for the malicious nodes to explore and launch various attacks for the same reason. Therefore, many researchers have designed secure multipath routing protocols \cite{3}.

However, as far as we know, none of the existing secure multipath routing protocols deals with the topology-exposure problem. Topology-exposure is a serious problem for MANET, which makes it possible for the malicious nodes to launch many kinds of attacks, such as black hole attack \cite{4}, wormhole attack \cite{5}, rushing attack \cite{6,7} and sybil attack \cite{8}. Topology-exposure is much more serious in multipath routing protocols than in other routing protocols considering that multipath routing protocols usually carry a lot of routing information in route messages in order to find sufficient routes. In some cases, data packets are also required to carry routing information. For example, Dynamic Routing Protocol (DSR) carries the whole route from source to destination in packet headers \cite{9}. Malicious nodes can deduce part or the whole network topology based on the captured routing information and it is hard to ensure the confidentiality of routing information because of the open media network environment in which any node can capture packets within its transmission range.

To deal with the topology-exposure problem, this paper thoroughly analyzes the threats brought by topology-exposure, defines topology-hiding and designs a TOpology-HIding multipath routing Protocol (TOHIP). TOHIP does not contain link connectivity information in route...
messages. Thus no node can deduce network topology by capturing route messages and the network topology is hidden. TOHIP can also find as many node-disjoint routes as possible, defend against attacks and exclude the unreliable routes. We formally prove that TOHIP is loop-free and topology-hiding. We also conduct intensive performance evaluation, which shows that TOHIP has better capability of finding routes and does not downgrade performance when there is no malicious node. When there are malicious nodes, TOHIP can greatly improve the packet delivery ratio at a low routing overhead and short routing convergent time.

The rest of this paper is organized as follows. Section 2 presents the threats of topology-exposure and defines topology-hiding. Section 3 discusses related works. Section 4 describes the design of TOHIP. Formal proof of the protocol’s characteristics and security analysis are given in Section 5. An enhanced protocol is proposed in Section 6, and performance evaluation is conducted in Section 7. Section 8 concludes this paper.

2. Topology-exposure problem and definition of topology-hiding

Consider an example MANET, whose topology is shown in Fig. 1. S is the source node and D is the destination node. There are two routes from node S to node D, which are $S \rightarrow C \rightarrow F \rightarrow D$ and $S \rightarrow A \rightarrow D$, in some multipath routing protocols. Based on the two routes, node D can conclude that $S$ is connected to A and C, C is connected to F, A and F are connected to D. Obviously, the two routes enable node D to obtain the whole network topology. We call this problem topology-exposure.

The knowledge of the network topology enables many kinds of attacks to be more harmful in MANET. Some examples are shown in Table 1. Taking the black hole attack as an example, if the malicious node intends to intercept the data packets to a specific destination, it should advertise that it has the route to this destination. It is difficult for this malicious node to redirect routing if no routing information is carried in packets. Also in order to choose the victim node and to intrude into a network, the malicious node needs to know network topology; otherwise, the malicious node cannot perform the black hole attack effectively. In addition to the attacks listed in Table 1, the launch of some other attacks, such as middle-person attack [10] and routing loops, also require the knowledge of network topology.

We use simulations to show the damage enabled by topology exposure. We take Secure Routing Protocol (SRP), a typical secure multipath routing protocol [11–13], as an example and study the effect when there are malicious nodes. Malicious Dropping Ratio (MDR) is defined to evaluate the bad effect brought by the malicious nodes.

$$MDR = \frac{\sum \text{data packet discarded by the malicious nodes}}{\sum \text{data packet sent by the source node}}$$

Two kinds of attacks, the black hole attack and the rushing attack, are launched in the simulation. The malicious nodes that launch black hole attack simply drop all packets passing by. When launching rushing attack, the malicious nodes not only get time advantage in route discovery by closing radio shock [6], but also drop all packets passing by. Fig. 2 shows the threat of topology-exposure by comparing the $MDR$ of the two attacks in MANET under different intrusion positions. From Fig. 2(a) we can see that when the attackers randomly choose positions to intrude into the network, the two kinds of attackers almost have the same $MDR$, which means they cause the same damage to the network. From Fig. 2(b) and (c), we can see the attackers in central positions can drop more packets than those in random positions. This is because the malicious nodes in the central position are likely to be included into the routes, so they can drop more packets. The simulation results show that the attackers that know network topology can give more damage to MANETs.

Both analysis and simulation show that some common attacks in MANET greatly leverage the knowledge of network topology. Hiding network topology can prevent many common attacks from the beginning and thus improve MANET security effectively. We define topology-hiding as follows. We define topology-hiding as follows.

**Definition 1.** Let $N$ be the set of all nodes in a MANET. Let $dist(n_i, n_j)$ be the hop count between a node $n_i$ and a node $n_j$. A routing protocol is topology-hiding only if:

- For any $n_i \in N$ and $n_j \in N$, if $dist(n_i, n_j) > 2$, then node $n_i$ cannot know which nodes are connected to node $n_j$.

<table>
<thead>
<tr>
<th>Name of attack</th>
<th>Principle of attack</th>
<th>Dependence on network topology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black hole [4]</td>
<td>Disrupt route discovery by redirecting routes</td>
<td>Choose the central positions to intrude into MANET</td>
</tr>
<tr>
<td>Wormhole [5]</td>
<td>Disrupt route discovery by using tunnel to reduce the hop count</td>
<td></td>
</tr>
<tr>
<td>Rushing [6,7]</td>
<td>Disrupt routing discovery by illegally getting the time advantage to forward route messages</td>
<td></td>
</tr>
<tr>
<td>Sybil [8]</td>
<td>Disrupt route discovery by disguising other nodes</td>
<td>Acquire other nodes’ identities</td>
</tr>
</tbody>
</table>

Fig. 1. Topology-exposure by routing information.

![Fig. 1](image-url)
In other words, topology-hiding is the requirement that any node can only deduce network topology within two hops at most.

### 3. Related works

Many of the existing multipath routing protocols have been derived from DSR [9] or AODV [14]. DSR-based protocols include ADSR [15], LD-DSR [16], MM-DSR [17], WI-DSR [18] and EMP-DSR [19]. Since DSR requires packet headers to carry routing information from source to destination, these protocols cannot hide topology [20]. Well-known AODV-based protocols include BAODV [21], IAODV [22], NDMR [23], AODVM [24] and AODV-BR [25]. Though AODV itself does not require routing information to be written in route messages, these protocols usually extend route messages to contain routing information so that they can find as many routes as possible. Thus, these protocols have the risk of exposing topology.

In addition, there is another kind of routing protocol, called the geographic routing [26,27]. In these protocols, each node learns its location through some localization techniques or location services. The malicious nodes can utilize the location information to deduce network topology. Therefore the geographic routing protocols may also expose network topology.

To combat attacks, numerous secure multipath protocols have been proposed [28–31]. These protocols usually emphasize on one or a portion of the five security requirements: confidentiality, integrity, availability, authentication and non-reputation, instead of hiding topology. Also, some of them are designed to defend against a particular kind of attack. For example, SAODV is effective in resisting the black hole attack but fails to detect the wormhole attack [28]. Some of them may work well in the presence of one malicious node, but become less effective in the presence of multiple colluding malicious nodes.

The anonymous routing protocols in MANETs, such as ALERT [32] and ALARM [33], can hide node identities from outside observers. However, they do not consider the topology-exposure problem. As far as we know, none of the existing multipath protocols and the countermeasures against attacks copes with the topology-exposure problem. To the best of our knowledge, we are the first one to point out the problem of topology exposure and to employ the idea of hiding topology to defend against attacks in MANET.

### 4. Protocol design

This section presents protocol TOpology-HIding multi-path routing Protocol (TOHIP). There are three objectives in designing TOHIP: (1) the link connection information is hidden as much as possible in route messages, so that the malicious nodes cannot deduce network topology; (2) even with prerequisite of hiding topology, TOHIP can find as many node-disjoint routes as possible, such that both load balancing and reliable packet delivery can be achieved; (3) TOHIP can exclude the malicious nodes from routes and detect the unreliable routes before transmitting packets. To achieve the goals, TOHIP employs the following mechanisms.

- Hide topology: TOHIP does not contain link connectivity information in route messages. Thus no node can deduce network topology by capturing route messages.
- Once a route is established, TOHIP will advertise a set containing the nodes that have been placed on routes, which prevents a node from being placed on another route.
- Defend against attacks: TOHIP uses the combination of hop count and round-trip time as routing metrics. Thus neither single wormhole attack nor single rushing attack can disrupt route discovery.
Examine the unreliable routes: TOHIP detects and excludes the unreliable routes by means of application-layer route probe messages before transmitting packets.

4.1. Overview and data structure

TOHIP has three phases: Route Request Phase, Route Reply Phase and Route Probe Phase.

- **Route Request Phase** creates reverse routes that will be used in Route Reply Phase. A route request message is transmitted from the source node to the destination node via broadcasting. To maintain network connectivity, upon receiving a route request message, every intermediate node creates a reverse route, and rebroadcasts the message if it has never received this message before.

- **Route Reply Phase** finds as many node-disjoint routes as possible. A route reply message is transmitted from the destination node to the source node via broadcasting. Upon receiving such a message, an intermediate node selects the neighbor closest to the source node as the previous hop on the route. It then advertises this selection to all its other neighbors, to ensure no node is selected on multiple routes.

- **Route Probe Phase** detects the unreliable routes before transmitting packets. To ensure the routes created in Route Reply Phase are available, the source node sends a route probe message through every discovered route to the destination node. By doing so, the unreliable routes can be detected and eliminated.

In the three phases above, no routing information is carried in route messages. In terms of data structure, every node keeps two tables. One is *Sequence Number Table (SNT)* which prevents nodes from rebroadcasting unnecessary route request messages. Each entry in SNT contains the source node which initially requests route discovery and the sequence number that the source node uses in this route discovery attempt. The other is *Routing Table (RT)*. Each entry in RT includes the destination node, the node through which to reach the destination, and the number of hops to the destination node. The two tables and the associated notations are shown in Table 2.

4.2. Route Request Phase

Before we present Route Reply Phase, we introduce the format of route request message first. A route request message (RREQ) contains the following fields.

- **S**: source ID.
- **D**: destination ID.
- **seq**: sequence number, set by the source node. \((S, \text{seq})\) uniquely identifies a RREQ message. All RREQ messages with the same \((S, \text{seq})\) belong to a same route discovery.
- **hopCt**: hop count to the source node.

Next, we will illustrate the actions of the source node, the intermediate nodes and the destination node, in the Route Request Phase.

4.2.1. At source node S

When a source node S needs a route to destination D but cannot find a route in its routing table, S initiates Route Request Phase by broadcasting a RREQ\((S, D, \text{seq}, \text{hopCt})\).

4.2.2. At intermediate nodes

Every intermediate node receiving route request message checks \((S, \text{seq})\) in SNT to determine whether this is the first RREQ copy for this route discovery attempt. If yes, they record \((S, \text{seq})\) in SNT, increases hopCt by 1, and then rebroadcast the RREQ message. Instead, they process every received copy and record the reverse route to the source via the sender of this copy. This is to find as many reverse routes as possible, which will be used in Route Reply Phase. The work flow of an intermediate node is shown in Fig. 3.

Please note that, in TOHIP, the number of routing entries in the routing table of an intermediate node is equal to the number of its neighbors. That is, for each node, the size of the routing table as well as the computational complexity on building the routing table is only related to the number of one-hop neighbors the node has, and is not determined by the size of the network. Thus, this mechanism will not bring huge computational overhead to the nodes.

4.2.3. At destination node D

When the destination node receives the first RREQ copy, it initiates a timer \(T_D\) to collect the following copies. Destination D only accepts the copies that arrive before

---

**Table 2**

Data structures.

<table>
<thead>
<tr>
<th>Sequence Number Table (SNT)</th>
<th>Destination node (D)</th>
<th>Next hop (nextHop)</th>
<th>Hop count (hopCt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source node (S)</td>
<td>seq</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Routing Table (RT)**

<table>
<thead>
<tr>
<th>Dest. node (D)</th>
<th>nextHop</th>
<th>hopCt</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Sequence Number Table (SNT)**

<table>
<thead>
<tr>
<th>Source node (S)</th>
<th>seq</th>
</tr>
</thead>
<tbody>
<tr>
<td>S#</td>
<td></td>
</tr>
</tbody>
</table>

---

![Fig. 3. Actions of an intermediate node after receiving a RREQ message.](image-url)
times out. It processes them in the same way as the intermediate nodes do, but does not rebroadcast.

### 4.3. Route Reply Phase

A route reply message (RREP) contains the following fields.

- S: source ID.
- D: destination ID.
- hopCt: hop count to the destination node.
- nextNode: the node through which the sender of the RREP message can reach the source node in the least number of hops. The field is to help the source node find multiple shortest routes to the destination, instead of many not-so-good routes.
- exNodeSet: is a set of nodes that cannot be intermediate nodes in the routes. This field is to ensure the routes found for the source are node-disjoint.

Next, we will illustrate present the actions of the destination node, the intermediate nodes and the source node, in the Route Reply Phase.

#### 4.3.1. At destination node D

Route Reply Phase is initiated by destination D to establish multiple node-disjoint routes from source S to destination D. Waiting for a certain period of time after destination D receives the first RREQ copy, it initiates Route Reply Phase by broadcasting a RREP message, in which hopCt is 0, exNodeSet contains all neighbors of destination D and nextNode is NULL. To set the node exclusion list exNodeSet to be all the neighbors of D is counter intuitive. The philosophy behind it is if D can be reached through a direct neighbor, a route with a detour through two intermediate neighbors should be avoided.

#### 4.3.2. At intermediate nodes

When an intermediate node \( n_i \) receives a RREP copy, it takes several actions. The first action is to prune its routing table based on the received information. \( n_i \) removes all the routes whose destination is the source node and whose Next Hop is in exNodeSet of the RREP. The action is to remove all the routes which use some nodes on an already established route and ensure all the established routes are node-disjoint.

Only in two cases, node \( n_i \) takes additional actions. The first case is nextNode is \( n_i \) itself, which means the RREP sender has selected \( n_i \) as the next hop to the source node or the previous node to the destination on the route. The second case is nextNode is NULL, which means this RREQ comes from destination D and \( n_i \) is a direct neighbor of D. Only in these two cases, \( n_i \) is on an established route to the destination through the RREP sender or to the destination directly from the source.

In the above two cases, \( n_i \) needs to take additional actions as follows. Firstly, \( n_i \) creates a route to destination D via the sender of the RREP. Secondly, \( n_i \) finds the closest neighbor \( n_j \) to source S by checking its routing table, which will be placed on route as the previous node and be filled in nextNode field in the RREP to be rebroadcasted. Then, \( n_i \) removes all the other routes except the one that is closest to source S.

In addition, \( n_i \) updates and rebroadcasts the RREP message. \( n_j \) sets nextNode to be the closest neighbor \( n_j \), inserts it into exNodeSet, increases hopCt by 1, and then rebroadcasts the RREP message.

To help understanding the actions of intermediate nodes, we give some discussions as follows. In RREQ phase, every node builds a reverse route to the source via each of its neighbor in routing table. The reverse routes guarantee that the intermediate nodes can know the shortest path information to the source node when building a routing path from the source to the destination. If \( n_i \) cannot find the closest neighbor to source S, which means all its neighbors have been selected in routes by other intermediate nodes, it gives up the attempt to build a route. And other intermediate nodes’ attempts are not affected.

From the description above, we can see that:

- exNodeSet greatly reduces the probability that a node is placed on more than one routes.
- Every intermediate node independently makes routing decisions by checking the reverse routes.
- Only the nodes that are placed on routes needs to rebroadcast the RREP message.
- Only two routes in routing table remains finally, which means that the established routes are bidirectional.

#### 4.3.3. At source node S

The RREP message keeps getting rebroadcasted until it arrives at source S. Once source S receives the first copy, it initiates a timer \( T_D \) to collect the following copies. Source S only accepts the copies that arrive before \( T_D \) times out, and processes them as the intermediate nodes but does not rebroadcast them. When \( T_D \) times out, source S stops accepting RREP message and multiple node-disjoint routes to destination D are established.

The detailed routing protocol is shown in Algorithm 1.

**Algorithm 1.** Protocol at node \( n_i \)

**Notations:**
- SNT,RT: as defined previously
- mhop: temporary hop count of the best route
- R1: temporary best route to the source

1. Upon receiving RREQ(\( S,D,seq,hopCt \)) from \( n_j \):
   - if \( n_i == D \) then
     - set a timer \( T_D \)
     - /* enter Route Reply Phase upon timeout*/
     - return
   - end if
   - Insert \( \langle S,n_j,hopCt+1 \rangle \) into RT /* reverse route */
   - if \( \langle S,seq \rangle \) doesn’t exist in SNT then
     - Insert \( \langle S,seq \rangle \) into SNT
   - end if
   - if \( n_j == D \) then
     - Rebroadcast RREQ \( \langle S,D,seq,hopCt+1 \rangle \)
   - end if

(continued on next page)
(2) Upon receiving \( RREP(S, D, \text{seq}, \text{hopCt}, \text{exNodeSet}, \text{nextNode}) \) from \( n_i \):

for each route \( R \) in \( RT \) do
  if \( R.D == S \) and \( R.nextHop \in \text{exNodeSet} \) then
    Remove route \( R \)
  end if
end for

if \( \text{nextNode} == \text{NULL} \) or \( \text{nextNode} == n_i \) then
  Insert \( (D, n_i, \text{hopCt} + 1) \) into \( RT \)
  /*two temporary parameters*/
  Set \( mHop = 65,535 \) and \( R1 = \text{NULL} \)
  for each route \( R \) in \( RT \) do
    if \( R.D == S \) and \( R.hopCount < mHop \) then
      Set \( mHop = R.hopCount \) and \( R1 = R \)
    end if
  end for
end if

for each route \( R \) in \( RT \) do
  /* remove all reverse routes to source*/
  if \( R.D == S \) then
    Remove route \( R \)
  end if
end for

if \( R1 != \text{NULL} \) then
  Insert \( R1 \) into \( RT \)
  nextNode = \( R1.nextHop \)
  \( \text{exNodeSet} = (R1.nextHop) \cup \text{exNodeSet} \)
  Broadcast \( RREP(S, D, \text{seq}, \text{hopCt} + 1, \text{exNodeSet}, \text{nextNode}) \)
end if

(3) Upon Timeout at the Destination:
  Set \( \text{exNodeSet} mHop = \text{NULL} \)
  for each route \( R \) in \( RT \) do
    \( \text{exNodeSet} = (R.nextHop) \cup \text{exNodeSet} \)
  end for
  Broadcast \( RREP(S, D, 0, \text{exNodeSet}, \text{NULL}) \)

4.4. Route Probe Phase

The routes in MANET may become unreliable due to node movement or malicious nodes. Before transmitting packets, source \( S \) initiates Route Probe Phase by sending a route probe message (RPRO) to destination \( D \) through every route that has been established in Route Reply Phase. For each arrived RPRO, destination \( D \) is required to send a RPRO message back to source \( S \) through the reverse route.

Route Probe Phase serves two goals: (1) to detect the unreliable routes. If there are malicious nodes dropping packets on a route, source \( S \) may not receive the returning RPRO message on that route. Thus the unreliable routes can be detected by source \( S \); (2) to find the secure shortest route. Source \( S \) treats the route on which the first-returning RPRO is received as the shortest route. Considering the fact that hop count is used as routing metric in Route Reply Phase, there will not exist the wormhole attackers or rushing attackers on this route.

4.5. An example

In this section, we use a 11-node network to illustrate the whole process of our routing protocol. The network topology is shown in Fig. 4(a).

4.5.1. Action of source \( S \) in Route Request Phase

When source \( S \) wants to learn the routes to destination \( D \), it initiates Route Request Phase by broadcasting a RREQ message \( (S, D, \text{seq}, \text{hopCt} = 0) \).

4.5.2. Action of intermediate nodes in Route Request Phase

Taking node \( A \) as an example. As shown in Fig. 4(a), node \( A \) has five neighbor nodes: \( B, C, E, F \) and \( G \). After node \( A \) gets the first RREQ message \( (S, D, \text{seq}, \text{hopCt} = 1) \) from node \( G \), it inserts \( (S, \text{seq}) \) into its SNT, inserts \( (S, G, \text{hopCt} + 1) \) into its RT and rebroadcasts RREQ \( (S, D, \text{seq}, \text{hopCt} + 1) \). Node \( A \) may also receive RREQ copies from node \( B, C, E \) and \( F \).
each of these copies, node \( A \) only creates a reverse route via the sender of that copy, but does not rebroadcast it. Fig. 5 shows the action of node \( A \) in Route Request Phase after receiving the RREQ messages from its neighbor nodes. Other intermediate nodes do a similar job as node \( A \). After Route Request Phase, every intermediate node learns all the reverse routes to source \( S \) via their neighbor nodes, which is shown in Fig. 4(b).

4.5.3. Action of destination \( D \)

Certain time after destination \( D \) receives the first RREQ copy, it initiates Route Reply Phase by broadcasting a RREP message \( \langle S, D, \text{hopCt} = 0, \text{exNodeSet} = \{B, C, A\}, \text{nextNode} = \text{NULL} \rangle \).

4.5.4. Action of intermediate nodes in Route Reply Phase

Also taking node \( A \) as an example. In the example, node \( A \) receives a RREP message \( \langle S, D, \text{hopCt} = 1, \text{exNodeSet} = \{B, C, A\}, \text{nextNode} = A \rangle \) from node \( B \) (Fig. 6(a)). In the RREP message, exNodeSet specifies that node \( B, C \) and \( A \) itself have been excluded. Thus, node \( A \) removes the reverse routes: \( \langle S, E, 4 \rangle \) and \( \langle S, F, 3 \rangle \) (Fig. 6(b)). Also, the RREP message specifies node \( A \) is selected as the nextNode by node \( B \). Thus, \( A \) also needs to create a route to destination \( D \) through node \( B \). As shown in Fig. 6(c), node \( A \) creates a route to destination \( D \): \( \langle D, B, 2 \rangle \). In addition, by checking the three remained reverse routes \( \langle S, E, 4 \rangle, \langle S, F, 3 \rangle \) and \( \langle S, G, 2 \rangle \), node \( A \) selects the closest node to source \( S \), that is node \( G \), as the new nextNode. Then it removes \( \langle S, E, 4 \rangle \) and \( \langle S, F, 3 \rangle \) from RT. After that, as shown in Fig. 6(d), node \( A \) updates nextNode to be node \( G \), inserts it into exNodeSet, increases hopCt by 1, and then rebroadcasts the RREP message \( \langle S, D, \text{hopCt} = 2, \text{exNodeSet} = \{B, G, C, A\}, \text{nextNode} = G \rangle \). Finally, node \( A \) only needs to maintain two routes: \( \langle S, G, 2 \rangle \) and \( \langle D, B, 2 \rangle \). Thus our protocol is a bidirectional solution.

4.5.5. Action of source \( S \) in Route Reply Phase

Certain time after source \( S \) receives the first RREP copy, the route discovery will finish. Finally, two node-disjoint routes are created: \( S \rightarrow I \rightarrow F \rightarrow C \rightarrow D \) and \( S \rightarrow G \rightarrow A \rightarrow B \rightarrow D \) (Fig. 4(c)). Before transmitting packets, source \( S \) will initiate Route Probe Phase by sending a RPRO message along both routes. If there are malicious nodes dropping packets on some route, source \( S \) may not get the returning RPRO message on that route. Then the unreliable route is detected and excluded.

5. Protocol analysis

In this section, we first prove that TOHIP is loop-free, node-disjoint and topology-hiding. Then we analyze TOHIP's capability in resisting attacks.

5.1. Proof of the characteristics of TOHIP

**Theorem 1.** TOHIP is loop-free and node-disjoint.
Thus, the potential damages incurred by malicious nodes are greatly reduced or even eliminated. Next, we analyze the robustness of TOHIP in resisting the following attacks when the attacks are launched from random positions.

5.2.1. Black hole attack
A black hole attacker disrupts route discovery by forging a route to the destination. A typical attack is launched as follows: When Source S broadcasts a route request to search a route to destination D, attacker A replies and advertises a route $R_{0A}$ from itself to destination D. If source S sends packets to destination D via route $R_{0A}$, the attacker A can intercept and discard the packets. Since TOHIP does not allow intermediate nodes to send route reply messages, it can resist the black hole attack.

5.2.2. Wormhole attack
A typical wormhole attack is launched as follows. Two collaborating attackers first select two central positions in the network to reside such that they are located on many potential routes. Then they build a private tunnel between them and advertise a fake hop count which is smaller than the real hop count between them. The action disrupts the route discovery mechanisms which only use hop count as routing metric since the private channel between the two attackers will always be selected as part of routes considering the smaller hop account. TOHIP can resist wormhole attack because (1) it is topology-hiding and it is impossible for attackers to choose central positions to launch the attack and (2) it uses round-trip time as a routing metric in Route Probe Phase, which makes it robust against hop count modification.

5.2.3. Rushing attack
Rushing attack is one of the denial of service attacks. While a normal node waits for a random delay before sending a packet to avoid collision in wireless communication, a rushing attacker always forwards packets immediately. Because of the rush, the round trip time recorded by a route request is always smaller than the true value if an attacker is on the route, and therefore the route is likely to be selected as the shortest route. TOHIP uses hop count as a routing metric in Route Reply Phase, and thus is resistant to the rushing attack.

5.2.4. Sybil attack
A sybil attacker disrupts route discovery by impersonating multiple legal nodes. To launch this attack, the attacker first obtains the identity of a set of legal nodes and then impersonates some or all of them to participate in multiple route discoveries. TOHIP does not include identity nor topology information in the routing messages, and thus it is impossible for malicious nodes to obtain the identity information of other nodes. Therefore, TOHIP is resistant to sybil attack.

6. Enhanced TOHIP with neighbor authentication
We enhance TOHIP by integrating a neighbor authentication mechanism with it. The improved TOHIP can resist
more attacks. In the following, we first present the enhanced TOHIP and then analyze its robustness when facing attacks other than aforementioned ones.

Before joining MANET, every node obtains a certificate from a trusted certificate server $T$ using the method described in [34]. From this certificate, every node has a pair of public key $PK$ and private key $SK$, and it keeps its authentication information $IMSG$:

$$IMSG = \left[ t, ID, SK(\text{t.ID}), PK \right],$$

where $t$ is the lifetime of the authentication information. The information is used to authenticate the identity of the sender of a route message and verify the message’s integrity. A node always inserts its $IMSG$ into the message it initiates.

Before we present the Route Request Phase with neighbor authentication, we introduce the following notations: $S$ and $D$ represents the source node and the destination node, respectively. $A$ and $B$ represents two intermediate nodes, and $N$ represents a neighbor of the destination. $*$ indicates that the message is sent via broadcasting.

The Route Request Phase has multiple steps:

1. **Step 1**: $S*A$: $RREQ = [S, seq, D, hopCt, SK_S, IMSG_S]$
2. **Step 2**: $A*B$: $RREQ = [S, seq, D, hopCt, SK_A, IMSG_A]$
3. **Step 3**: $B*D$: $RREQ = [S, seq, D, hopCt, SK_B, IMSG_B]$

The Route Reply Phase also has multiple steps:

1. **Step 1**: $D*N$: $RREP = [S, D, nextNode, exNodeSet, SK_D, IMSG_D]$
2. **Step 2**: $N*B$: $RREP = [S, D, nextNode, exNodeSet, SK_N, IMSG_N]$
3. **Step 3**: $B*A$: $RREP = [S, D, nextNode, exNodeSet, SK_B, IMSG_B]$
4. **Step 4**: $A*S$: $RREP = [S, D, nextNode, exNodeSet, SK_A, IMSG_A]$

With neighbor authentication, before transmitting packets, the source node first verifies the availability of routes and finds the shortest secure route. Then it determines the transmission policy according to the mechanism in [34] based on the number of available routes. When the number of the available routes is less than or equal to three, the source node uses single route policy to forward packet; otherwise, multiple-route policy is applied. The source node restarts the route discovery if there is no available route.

The enhanced TOHIP can resist other attacks. For example, modification attack can be detected by authenticating the integrity of route messages. Impersonation attack can be prevented because every node is required to authenticate its neighbors. Fabrication attack can be defeated by appending a signature to route messages.

7. Performance evaluation

7.1. Simulation methodology

We implement enhanced TOHIP in NS-2 network simulator. Our objectives in conducting this evaluation are fourfold: firstly, evaluating the capability of TOHIP in finding routes; secondly, testing the effectiveness of TOHIP in delivering packets in both non-adversarial and adversarial scenarios; thirdly, checking the overhead of TOHIP; finally, studying the performance of TOHIP under different conditions, including the different number of attackers and node speed. Like many other multipath routing protocols, we choose SRP as the comparison scheme [12,13].

To evaluate the capability of finding routes, like the evaluation in [24], we employ the number of node-disjoint routes between the random pairs of source and destination discovered by the protocol as the evaluation metric. To evaluate effectiveness of delivering packets and the associated overhead, we formally define the following three metrics.

- **Packet Delivery Ratio (PDR)**: the ratio of packets successfully delivered to packets generated. More precisely, we are interested in the network layer PDR. We aim to capture the raw network performance in the presence of attackers, without using any packet retransmission scheme, either at network or upper layer.

$$PDR = \frac{\sum \text{packet received by the destination}}{\sum \text{packet generated by the source}} \times 100\%$$

- **Routing Overhead measured by Packets (ROP)**: the average number of route messages (in packets) per successfully delivered packet.

$$ROP = \frac{\sum \text{route message}}{\sum \text{data packet received by the destination}}$$

- **Routing Overhead measured by Bytes (ROB)**: the average number of route messages (in bytes) per successfully delivered packet.

$$ROB = \frac{\sum \text{size of route message}}{\sum \text{data packet received by the destination}}$$

- **End-to-End Delivery Delay (EED)**: the average end-to-end delay per successfully delivered packet.

$$EED = \frac{\sum \text{end-to-end delay for each packet}}{\sum \text{data packet received by the destination}}$$

In the simulation, mobile nodes follow the random waypoint mobility model. The channel capacity is 2 Mb/s and the maximum communication range is 250 m. Other parameters are listed in Table 3. All results shown are the average of 50 experiments.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation area</td>
<td>1000 m $\times$ 1000 m</td>
</tr>
<tr>
<td>Number of mobile nodes</td>
<td>50</td>
</tr>
<tr>
<td>Simulation time</td>
<td>800 s</td>
</tr>
<tr>
<td>Pause time</td>
<td>30 s</td>
</tr>
<tr>
<td>Number of source-destination pair</td>
<td>10</td>
</tr>
<tr>
<td>Packet generation rate</td>
<td>1 packet/s</td>
</tr>
<tr>
<td>Packet size</td>
<td>512 bytes</td>
</tr>
<tr>
<td>Node movement speed</td>
<td>[0.12 m/s]</td>
</tr>
<tr>
<td>Number of attackers</td>
<td>0–10</td>
</tr>
</tbody>
</table>
7.2. Capability of finding routes

Most existing multipath routing protocols [2,35] only build reversed route for the first received RREP message. We call OFC (Only First Copy). On the contrary, our protocol builds reversed routes of all the received RREP messages, so that it maintains rich network connectivity information in Route Request Phase. An example is illustrated in Fig. 8. Fig. 8(a) is the original topology. Fig. 8(b) and (c) show the reverse routes after Route Request Phase in our protocol and OFC, respectively. From Fig. 8(b), we can see our protocol maintains the network connectivity because it does not discard any RREQ messages. OFC loses much of the network connectivity information because it only builds route for the first received RREQ message. Fig. 8(d) and (e) show the established routes after Route Reply Phase.

To explain the TOHIP’s principle, we give its detailed procedure in RREP phase to find the multiple node-disjoint routes as follows.

- **Step 1**: destination D selects node 5, node 6, and node 7 as the upstream nodes (this rule is presented in Section 4.3). We have three partial node-disjoint routes: $5 \rightarrow D, 6 \rightarrow D, 7 \rightarrow D$.

- **Step 2**: node 5 will select node 1 as the upstream node because it is the node closest to source S among node 5’s neighboring nodes. Thus we have a partial route: $1 \rightarrow 5 \rightarrow D$.

- **Step 3**: node 6 will select node 4 as the upstream node. Although both node 4 and node 7 are node 6’s neighboring nodes and have the same hop count to source S, node 4 is selected because it is ahead of node 7 in the reverse routing table. Thus we have a partial route: $4 \rightarrow 6 \rightarrow D$.

- **Step 4**: node 7 will select node 3 as the upstream node. Thus we have a partial route: $3 \rightarrow 7 \rightarrow D$.

- **Step 5**: node 1 directly connects to source S. So we have a route: $S \rightarrow 1 \rightarrow 5 \rightarrow D$.

- **Step 6**: node 4 can only select node 2 as the upstream node because its neighbors, node 1 and node 3, have been selected by other nodes (in other words, they are included in exNodeSet). Thus we have a partial route: $2 \rightarrow 4 \rightarrow 6 \rightarrow D$.

- **Step 7**: node 3 directly connects to source S. So we have a route: $S \rightarrow 3 \rightarrow 7 \rightarrow D$.

- **Step 8**: node 2 directly connects to source S. So we have a route: $S \rightarrow 2 \rightarrow 4 \rightarrow 6 \rightarrow D$.

- Finally, we have three node-disjoint routes shown in Fig. 8(d).

![Fig. 8. Procedure of route discovery.](image-url)
In total, our protocol creates three node-disjoint routes: 
\[ S \rightarrow 1 \rightarrow 5 \rightarrow D, S \rightarrow 2 \rightarrow 4 \rightarrow 6 \rightarrow D \quad \text{and} \quad S \rightarrow 3 \rightarrow 7 \rightarrow D, \]
while OFC only creates two: 
\[ S \rightarrow 1 \rightarrow 5 \rightarrow D \quad \text{and} \quad S \rightarrow 2 \rightarrow 4 \rightarrow 7 \rightarrow D. \]

The route discovery capacity of TOHIP and SRP [24], a OFC-based protocol, is compared in Fig. 7. From the figure we can see that the capability of finding routes decreases as the minimum hop count between source and destination increases for both schemes. The reason is that the routes tend to intersect with each other as the hop count increases. From the figure, we can see that TOHIP outperforms SRP.

TOHIP can find six routes when the minimum hop count is 2, while SRP only finds 3.5 routes. TOHIP still can find two routes even when the minimum hop count is up to 6, while SRP only finds one route. TOHIP outperforms SRP because they adapt different mechanisms to deal with RREQ message in Route Request Phase. The intermediate nodes in TOHIP process all received RREQ message, and create a reverse route for each received copy. However, the intermediate nodes in SRP only accept the first arrived copy, which means an intermediate node only creates one reverse route. Thus it finds less routes in Route Reply Phase.

7.3. Non-adversarial scenario

Fig. 9 shows how the maximum speed of node movement affects the performance in aspects of packet delivery ratio, routing overhead and end-to-end delay in the non-adversarial scenario where there is no attacker.

- From Fig. 9(a), the packet delivery ratio decrease as the maximum speed increases. Both TOHIP and SRP keep the packet delivery ratio at more than 97%.
- From Fig. 9(b), the routing overhead increases as the maximum speed increases. Compared to SRP, TOHIP has a very similar routing overhead.
- From Fig. 9(c), TOHIP has more routing overhead than SRP. This is because RREP messages in TOHIP contains an exclusion node list (exNodeSet), which increase the size of the route message.
- From Fig. 9(d), both TOHIP and SRP have the end-to-end delay in the range of [0.030 s, 0.035 s]. Also this metric keeps relative stable, which proves that our protocol does not degrade the efficiency of delivering packets.

The simulation results above show our protocol does not degrade the performance. Our protocol achieves a very similar performance as SRP in the scenario where there is no attacker.

7.4. Adversarial scenario

Next we will evaluate the performance in the adversarial scenario when there are malicious nodes performing black hole attack and rushing attack.

7.4.1. Packet delivery ratio

Fig. 10(a) shows how the number of different attackers affect the packet delivery ratio in SRP. The rushing attackers drop more packets that the black hole attackers. This is because the rushing attackers have time advantage to forward route messages, and thus they are more likely...
to be placed on routes than the black hole attackers that only advertise the forged shorter routes.

Fig. 10(b) and (c) compare TOHIP with SRP when there are black hole attackers and rushing attackers, respectively. SRP is affected greatly by the attackers. The packet delivery ratio in SRP decreases to 60% as the number of attackers increases to 10. However, the number of attackers have little impact on TOHIP. The packet delivery ratio in TOHIP keeps stable at above 97% even there are 10 attackers. The results show that TOHIP can resist black hole attack and rushing attack effectively. This is because: (1) TOHIP can exclude the unreliable routes in Route Probe Phase before transmitting packets; (2) TOHIP uses hop count and round-trip time as routing metrics in Route Reply Phase and Route Probe Phase respectively, thus neither the single rushing attack nor the single wormhole attack can disrupt route discovery.

7.4.2. Routing overhead

Fig. 11 depicts the normalized routing overhead. When the number of attackers is up to 10, our protocol brings about 50% more routing overhead than SRP. However, our protocol improves the packet delivery ratio from 52.3% of SRP to 97.9% when there are 10 attackers. Our protocol incurs more routing overhead than SRP as a result of three reasons. The first one is TOHIP needs to detect the unreliable routes before transmitting packets. The second one is TOHIP needs to contain exNodeSet in RREP messages that guarantees the final routes node-disjoint. This field augments the size of route messages. The third one relates to the fact that in the presence of many attackers, the routes are more likely to become unreliable, thus TOHIP needs to invoke route discovery more often to find the fresher routes.

7.4.3. End-to-end delay

Fig. 12 depicts the end-to-end delay, which reflects the average transmission delay from source to destination. The end-to-end delay decreases slightly as the number of attackers increases. This is because the long latency packets are likely to be discarded as the number of attackers increases. Also this figure shows that TOHIP does not increase the end-to-end delay.

7.5. Convergent time

The routing convergent time in TOHIP depends on the timers (TS and TD) that are configured to collect RREQ
messages and RREP messages in Route Request Phase and Route Reply Phase, respectively. When the timers are set to be 0.3 s, the convergent time in TOHIP is 0.4 s, while it is 0.2 s in SRP. TOHIP has longer convergent time because: (1) TOHIP tends to establish longer routes to prevent them from intersecting with each other; (2) TOHIP tends to find as many node-disjoint routes as possible in a route discovery attempt to prevent route discovery from being invoked frequently.

8. Conclusion

After analyzing the common attacks and their dependence on the acquisition of network topology, this paper points out the necessity of hiding topology in designing the routing protocols in MANETs. The paper also formally defines topology-hiding and proposes a TOpology-Hiding multipath routing Protocol (TOHIP). Performance evaluation shows that TOHIP has better capability of finding routes. TOHIP does not degrade the performance when there is no attack. While in the adversarial scenario, the simulation results show that TOHIP can resist attacks at a low overhead and short routing convergent time. As for the future work, we plan to design the data transmission strategy with fault detection mechanism based on TOHIP.

Acknowledgement

This work is supported by National Program on Key Basic Research Project (2012CB315804), National Natural Science Foundation of China (60803139, 61173133, 61133015), and NSF Grant 1128369.

References


Yujun Zhang is an Professor in the Institute of Computing Technology, Chinese Academy of Sciences. He received his B.S. degree and Ph.D. degree from Nankai University in 1999 and Chinese Academy of Sciences in 2004, respectively. He is author or coauthor of more than 50 journal and conference papers. He also has fifteen patents in China and has published one book on trusted network protocol. From 2009 to 2010, he was a visiting scholar at University of California, Los Angeles (UCLA), USA. He has served in technical committees of several national and international conferences. His current research interests include mobile network, network protocol and architecture.

Tan Yan received his PhD in 2014, under the supervision of Dr. Guiling Wang, in Department of Computer Science, New Jersey Institute of Technology. He obtained M.S. from Department of Electrical & Computer Engineering, NJIT in 2009, and B.E. in 2007 from School of Information Science and Technology, Southeast University, Nanjing, China.

Guiling Wang joined NJIT in fall 2006 and was promoted to an Associate Professor with tenure in June 2011. She received her Ph.D. in Computer Science and Engineering and a minor in Statistics from The Pennsylvania State University in May 2006. She received her B.S. in Software from Nankai University in Tianjin, China.

Zhongcheng Li received his B.S. degree in computer science from Peking University in 1983, and M.S. and Ph.D. degrees from Institute of Computing Technology, Chinese Academy of Sciences in 1986 and 1991, respectively. From 1996 to 1997, he was a visiting professor at University of California at Berkeley. He is currently a professor of Institute of Computing Technology, Chinese Academy of Sciences. His current research interests include next generation Internet, dependable systems and networks.