Design and Performance Study of a Topology-Hiding Multipath Routing Protocol for Mobile Ad Hoc Networks

Yujun Zhang[†], Guiling Wang[‡], Qi Hu[†], Zhongcheng Li[†], Jie Tian[‡]
[†]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, P.R.China
[‡]Department of Computer Science, New Jersey Institute of Technology, Newark, USA
Email: {zhmj,huqi,zcli}@ict.ac.cn, {gwang,jt66}@njit.edu

***Abstract*—Existing multipath routing protocols for MANET ignore the topology-exposure problem. This paper analyzes the threat of topology-exposure and proposes a Topology-Hiding Multipath Routing protocol (THMR). THMR doesn't allow packets to carry routing information, so malicious nodes cannot deduce topology information 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 THMR is loop-free and topology-hiding. Simulation results show that our protocol has better capability of finding routes and can greatly increase the capability of delivering packets in the scenario where there are attackers at the cost of low routing overhead.**

I. 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 [1], [2]. However, multipath routing protocols may become a vulnerable target for malicious nodes to explore and launch various attacks for the same reason. Therefore, many researchers have designed secure multipath routing protocols [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 malicious nodes to launch many kinds of attacks, such as black hole attack [4], wormhole attack [5], rushing attack [6], [7] and sybil attack [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, the Dynamic Routing Protocol (DSR) carries routing information from source to destination in packet headers [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 (THMR). THMR does not contain link connectivity information in route messages. Thus no node can deduce network topology by capturing route messages and the topology is hidden. THMR can also find as many node-disjoint routes as possible, defend against attacks and exclude unreliable routes. We formally prove that THMR is loop-free and topology-hiding. We also conduct intensive performance evaluation, which shows that THMR has better capability of finding routes and doesn't downgrade performance when there is no malicious node. When there are malicious nodes, THMR can greatly improve the packet delivery ratio at a low overhead and short routing convergent time.

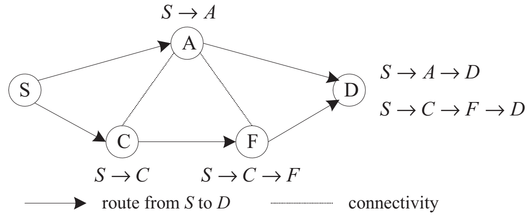


Fig. 1. Topology-exposure by routing information

The rest of this paper is organized as follows. Section II presents the threats of topology-exposure and defines topology-hiding. Section III discusses related works. Section IV describes the design of THMR. Formal proof of the protocol's characteristics is shown in section V and performance evaluation is conducted in section VI. Section VII concludes this paper.

II. 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 $S \to C \to F \to D$ and $S \to A \to D$, in some multiple routing protocols. Based on the two routes, node D can conclude that S is connected to A and C, C is connected to F and F is connected to D. Obviously, the two routes enable node D to obtain the whole network topology. We call this problem topology-exposure.

978-1-4673-0775-8/12/$31.00 ©2012 IEEE

The knowledge of the network topology enables many kinds of attacks to be more harmful in MANET. Some examples are shown in Table I. 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 I, the launch of some other attacks, such as middleperson attack [10] and routing loops, also require the knowledge of network topology.

TABLE I
ATTACKS VS. NETWORK TOPOLOGY

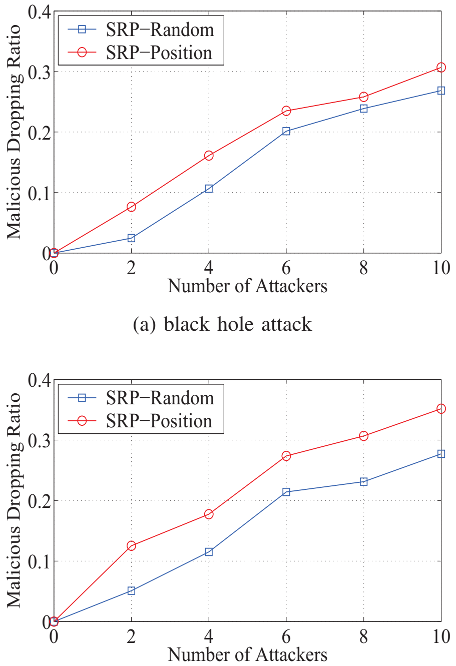
Name of attack	*Principle of attack*	*Dependence on network topology*
Black hole [4]	Disrupt route discovery by redirecting routing.	Choose the central position to intrude into MANET.
Wormhole [5]	Disrupt route discovery by using tunnel to reduce the hop count.	
Rushing [6], [7]	Disrupt routing discovery by illegally getting the time advantage to forward route messages.	
Sybil [8]	Disrupt route discovery by imitating other node.	Acquire other nodes' identities.

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 presented in SRP. We call the scenario *SRP-Random* when the malicious nodes randomly choose positions to intrude into the MANET, and the scenario *SRP-Position* when the malicious nodes choose the central position to intrude into the MANET. *Malicious Dropping Ratio* ($\overline{MDR}$) is defined to evaluate the bad effect of the malicious nodes.

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

Two kinds of attacks, the black hole attack and the rushing attack, are launched in *SRP-Random* and *SRP-Position*. 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, but also drop all packets passing by. From Fig. 2(a) and Fig. 2(b), the malicious dropping ratio of *SRP-Position* is obviously higher than *SRP-Random* in both black hole attack scenario and rushing attack scenario. 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 malicious nodes that know network topology can give more damage to MANET.

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.



(a) black hole attack

(b) rushing attack

Fig. 2. Malicious dropping ratio in *SRP-Random* and *SRP-Position*

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.

In other words, topology-hiding is the requirement that any node can only deduce network topology within two hops at most.

III. 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 header to carry the route to the 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 doesn't require routing information to be written in route messages, these protocols usually extend route messages to contain routing information so that they can establish as many routes as possible. Thus, these protocols have the risk of exposing topology.

In addition, there is another kind of routing protocols, 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 routing protocols have been proposed [3], [28], [29]. 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 [3]. 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.

As far as we knonw, 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.

IV. Protocol Design

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

- Hide topology: THMR does't contain link connectivity information in route messages. Thus no node can deduce network topology by capturing route messages.
- Find node-disjoint routes: Once a route is established, THMR 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: THMR 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.
- Exclude unreliable routes: THMR detects and excludes unreliable routes by means of application-layer route probe messages before transmitting packets.

A. Overview and Data Structure

THMR has three phases: *Route Request Phase*, *Route Reply Phase* and *Route Probe Phase*. In these phases, no routing information is carried in route messages. In *Route Request Phase*, the source node broadcasts a route request message. Every intermediate node creates a reverse route to the source node for every received copy, but only rebroadcasts the first copy. After *Route Request Phase*, every node can establish multiple reverse routes back to the source node. This is to facilitate the discovery of multiple node-disjoint routes for the source in *Route Reply Phase*. In *Route Reply Phase*, a node-excluding mechanism is designed to find as many node-disjoint routes as possible. *Route Probe Phase* is to detect the malicious nodes and ensure the availability of the candidate routes before transmitting packets.

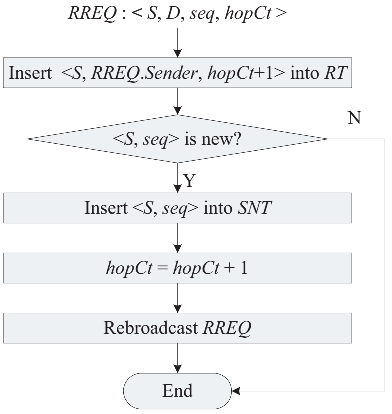


Fig. 3. Actions of node *A* after receiving a *RREQ* message

In terms of data structure, every node keeps two tables. One is the *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 the *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 II.

TABLE II
Data Structures

Source (S)	*Sequence number (seq)*
···	···

Sequence Number Table (*SNT*)

Destination (D)	*Next hop (nextHop)*	*Hop count (hopCt)*
···	···	···

Routing Table (*RT*)

B. Route Request Phase

Before we present *Route Reply Phase*, we introduce the format of route request message first. A route request message (*RREQ*) contains source ID (*S*), destination ID (*D*), a sequence number (*seq*) and distance to source (*hopCt*). *hopCt* is the distance to the source node. The sequence number is set by the source node. $< S, seq >$ uniquely identifies a *RREQ* message. All *RREQ* messages with the same $< S, seq >$ belong to a same route discovery.

When a source node *S* needs a route to a destination *D* but cannot find a route in its routing table, *S* initiates *Route Request Phase* by broadcasting $RREQ < S, D, seq, hopCt >$. Every intermediate node receiving the message checks $< S, seq >$ in *SNT* to determine whether this is the first *RREQ* copy for this route discovery attempt. If yes, they record $< S, seq >$ in *SNT*, increase *hopCt* by 1, and then rebroadcast the 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.

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 T_D times out, and processes them as the intermediate nodes do but does not rebroadcast them.

C. Route Reply Phase

A route reply message (*RREP*) contains the source ID (*S*), destination ID (*D*), the distance to the destination (*hopCt*) and two very important fields: *nextNode* and *exNodeSet*. *nextNode* is the node through which the sender of the *RREP* message can reach the source node in the least number of hops. The filed is to help the source node find multiple shortest routes to the destination, instead of many not-so-good routes. *exNodeSet* is a set and contains all the nodes that cannot be one intermediate node in the routes. This field ensures that the multiple routes found for the source are node-disjoint.

■ At destination *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.

■ 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 in the *RREP* 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 further do some processing task as follows. Firstly, n_i creates a route to destination *D* through the *RREP* sender. 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

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 $< S, n_j, hopCt + 1 >$ into RT /* reverse route */
if $< S, seq >$ doesn't exist in SNT then
  Insert $< S, seq >$ into SNT
  if $n_i != D$ then
    Rebroadcast $RREQ < S, D, seq, hopCt + 1 >$
  end if
end if

(2) Upon receiving $RREP < S, D, seq, hopCt,$
$exNodeSet, nextNode >$ from $n_j$:
for each route R in RT do
  if $R.D == S$ and $R.nextHop \in exNodeSet$ then
    Remove route R
  end if
end for
if $nextNode == NULL$ or $nextNode == n_i$ then
  Insert $< D, n_j, hopCt + 1 >$ into RT
  /* two temporary parameters */
  Set $mHop = 65535$ and $R1 = NULL$
  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
  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 $R1! = NULL$ then
    Insert R1 into RT
    $nextNode = R1.nextHop$
    $exNodeSet = \{R1.nextHop\} \cup exNodeSet$
    Broadcast $RREP < S, D, seq, hopCt + 1,$
    $exNodeSet, nextNode >$
  end if
end if

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

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_i sets *nextNode* to be the closest neighbor n_j, inserts it into *exNodeSet*, increases *hopCt* by 1, and then rebroadcasts the *RREP* message.

From the description above, we can see that

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

■ At the source node: The *RREP* message keeps getting rebroadcasted until it arrives at source *S*. Once source *S* receives the first copy, it initiates a timer T_S to collect the following copies. Source *S* only accepts the copies that arrive before T_S times out, and processes them as the intermediate nodes but does not rebroadcast them. When T_S 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.

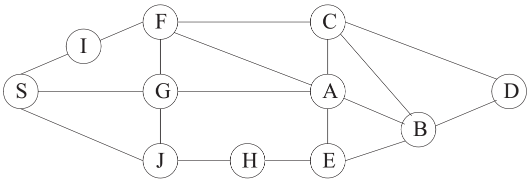


Fig. 4. Original topology

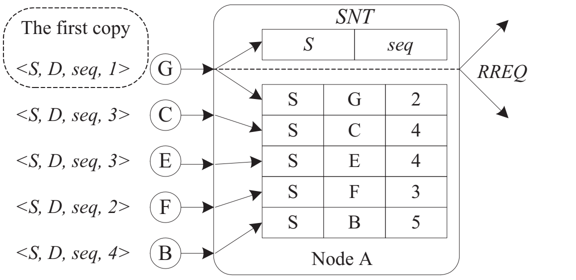


Fig. 5. Action of intermediate node (node *A*) in *Route Request Phase*

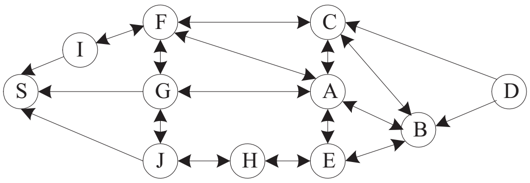


Fig. 6. Reverse routes after *Route Request Phase*

D. 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 is malicious node 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 won't be an attacker performing wormhole attack or rushing attack on this route.

E. 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.

■ Action of source *S* in *Route Request Phase*:

When source *S* wants to learn the routes to destination *D*, it initiates the *Route Request Phase* by broadcasting a *RREQ* message $< S, D, seq, hopCt = 0 >$.

■ Action of intermediate nodes in *Route Request Phase*:

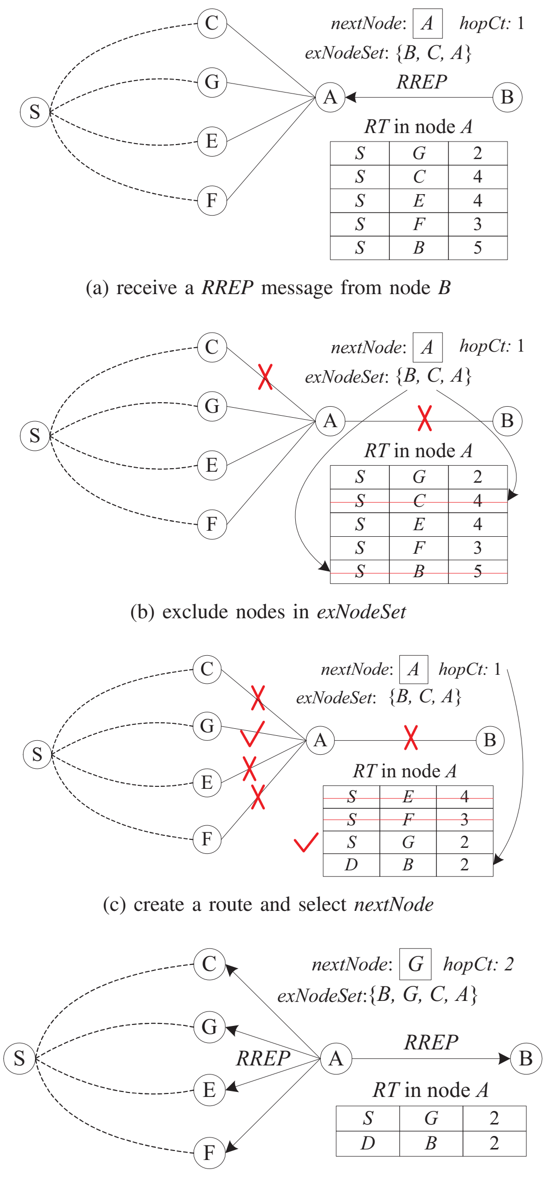
Taking node *A* as an example. As shown in Fig. 4, node *A* has five neighbor nodes: *B*, *C*, *E*, *F* and *G*. After node *A* gets the first *RREQ* message $< S, D, hopCt >$ from node *G*, it inserts $< S, seq >$ into its *SNT*, inserts $< S, G, hopCt + 1 >$ into its *RT* and rebroadcasts *RREQ* $< S, D, seq, hopCt + 1 >$. Node *A* may also receive *RREQ* messages from node *B*, *C*, *E* and *F*. For each of these messages, node *A* only creates a reverse route through the sender of that message, but does not rebroadcast it. Fig. 5 shows the action of node *A* in *Route Request Phase* after receives 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* through their neighbor nodes, which is shown in Fig. 6.

■ Action of destination *D*:

Certain time after destination *D* receives the first *RREQ* copy, it initiates *Route Reply Phase* by broadcasting a *RREP* message $< S, D, hopCt = 0, exNodeSet = \{B, C\}, nextNode = NULL >$.

■ Action of intermediate nodes in *Route Reply Phase*:

Also taking node A as an example. In the example, node *A* receives a *RREP* message $< S, D, hopCt = 1, exNodeSet = \{B, C, A\}, nextNode = A >$ from node *B* (Fig. 7(a)). In the *RREP* message, $exNodeSet$ specifies that node *B*, *C* and *A* itself have been excluded. Thus, node *A* removes the reverse routes: $< S, B, 5 >$, and $< S, C, 4 >$ (Fig. 7(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



(a) receive a *RREP* message from node *B*

(b) exclude nodes in *exNodeSet*

(c) create a route and select *nextNode*

(d) update and rebroadcast *RREP*

Fig. 7. Action of intermediate node (node *A*) in *Route Reply Phase*

D through node *B*. As shown in Fig. 7(c), node *A* creates a route to destination *D*: $< D, B, 2 >$. In addition, by checking the three remained reverse routes $< S, E, 4 >$, $< S, F, 3 >$ and $< S, G, 2 >$, node *A* selects the closest node to source *S*, that is node *G*, as the new *nextNode*. Then it removes $< S, E, 4 >$ and $< S, F, 3 >$ from *RT*. After that, as shown in Fig. 7(d), node *A* updates *nextNode* to be node *G*, inserts it into *exNodeSet*, increases *hopCt* by 1, and then rebroadcasts the *RREP* message $< S, D, hopCt = 2, exNodeSet = \{B, G, C, A\}, nextNode = G >$. Finally, node *A* only needs to maintain two routes: $< S, G, 2 >$ and $< D, B, 2 >$. Thus our protocol is a bidirectional solution.

■ 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. 8). 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.

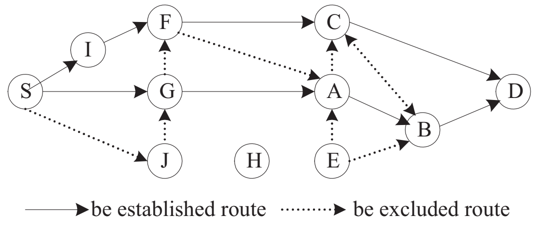


Fig. 8. Node-disjoint routes after *Route Reply Phase*

V. Protocol Analysis

Theorem 1. *THMR is loop-free and node-disjoint.*

Proof: In *Route Request Phase*, our protocol use $< S, seq >$ as the unique ID of *RREQ* messages, which guarantees that no *RREQ* message will be forwarded twice.

In *Route Reply Phase*, once a node is placed on a route, it will be inserted into *exNodeSet*. The nodes in *exNodeSet* cannot be placed on any route again.

Hence, THMR is loop-free and node-disjoint. ■

Theorem 2. *THMR is topology-hiding.*

Proof: In *Route Request Phase*, no routing information is carried in *RREQ* message. Every node only knows its neighbors, and thus this meets the requirement of topology-hiding.

In *Route Reply Phase*, assume node n_i receives a *RREP* message from node n_k, where *exNodeSet*=$< n_m, n_{m+1}, \cdots, n_{m+n} >$ and *nextNode*=n_t. No node can deduce the topological relationship from this orderless set $< n_m, n_{m+1}, \cdots, n_{m+n} >$. As for *nextNode*, there are two cases: either it is NULL or it is a node n_t.

When *nextNode* is NULL, this *RREP* message must come from destination *D*. Thus n_i can only know that the nodes in *exNodeSet* are the neighbors of destination *D*. For any $n_j \in$ *exNodeSet*, there must exist $dist(n_i, n_j) \leq 2$ because both nodes are neighbors of destination *D*.

When *nextNode* is a node n_t, the sender n_k of the *RREP* message must have selected n_t as the next hop on the shortest route to the source. Thus n_i can deduce that n_k is a neighbor of n_t. Since n_i is also a neighbor of n_k, there must exist $dist(n_i, n_t) \leq 2$.

Considering the above two cases, we can conclude that for any node n_i and node n_j, if n_i can deduce a node is connected to n_j, there must exist $dist(n_i, d_j) \leq 2$. n_i cannot obtain any topology information more than that.

Hence, THMR is topology-hiding. ■

VI. Performance Evaluation

A. Simulation Methodology

THRM is implemented in NS-2 network simulator. Our objectives in conducting this evaluation are four-fold: firstly, evaluating the capability of THMR in finding routes; secondly, testing the effectiveness of THMR in delivering packets in both non-adversarial and adversarial scenarios; thirdly, checking the overhead of THMR; finally, studying the performance of THMR under different conditions, including the 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 delievering packets and the associated overhead, we formally define the following three metrics.

- Packet Delivery Ratio ($\overline{PDR}$): the ratio of packets successfully delivered to packets generated. More precisely, we are interested in the network layer $\overline{PDR}$. 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.

$$\overline{PDR} = \frac{\sum \text{data packet received by the destination}}{\sum \text{data packet generated by the source}}$$

- Routing Overhead (RO): the average number of route messages per successfully delivered packet.

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

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

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

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

TABLE III
DEFINITION OF TOPOLOGY-HIDING

Parameter	Values
Simulation area	$1000m \times 1000m$
Number of mobile nodes	50
Simulation time	$800s$
Pause time	$30s$
Number of source-destination pair	10
Packet generation rate	$1packet/second$
Packet size	$512byte$
Node movement speed	$[0, 12m/s]$
Number of attackers	$0 - 10$

B. Capability of finding routes

The route discovery capacity is shown in Fig. 9. 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 THMR outperforms SRP.

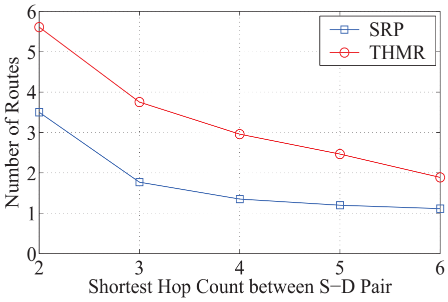


Fig. 9. Capability of finding routes

THMR can find six routes when the minimum hop count is 2, while SRP only finds 3.5 routes. THMR still can find two routes even when the minimum hop count is up to 6, while SRP only finds one route. THMR outperforms SRP because they adapt different mechanisms to deal with *RREQ* message in *Route Request Phase*. The intermediate nodes in THMR 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*.

C. Non-adversarial Scenario

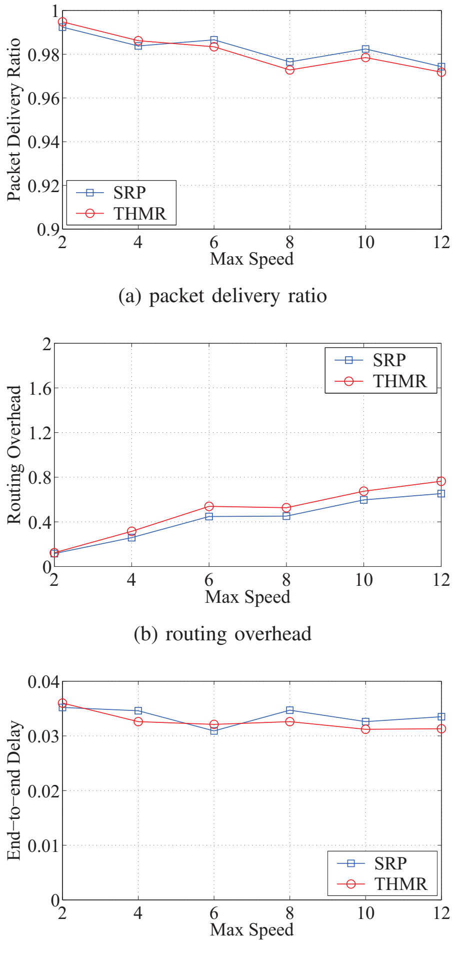
Fig. 10 shows how the maximum speed of node movement affects the performance in aspects of packet delivery ratio, routing overhead and end-to-end delay in the non-adversarial scenario where there is no attacker.

- From Fig. 10(a), the packet delivery ratio decrease as the maximum speed increases. Both THMR and SRP keep the packet delivery ratio at more than 97%.
- From Fig. 10(b), the routing overhead increases as the maximum speed increases. Compared to SRP, THMR has a very similar routing overhead.
- From Fig. 10(c), both THMR and SRP have the end-to-end delay in the range of $[0.030s, 0.035s]$. Also this metric keeps relative stable, which proves that our protocol doesn't degrade the efficiency of delivering packets.

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

D. Adversarial Scenario

Next we will evaluate the performance in the adversarial scenario when there are malicious nodes performing black hole attack and rushing attack. Fig. 11(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.

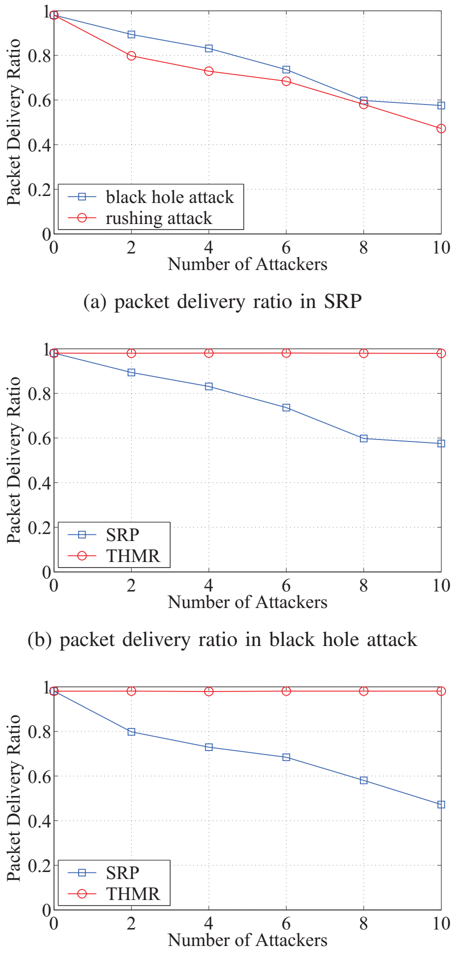


(a) packet delivery ratio

(b) routing overhead

(c) end-to-end delay

Fig. 10. Performance in non-adversarial scenario



(a) packet delivery ratio in SRP

(b) packet delivery ratio in black hole attack

(c) packet delivery ratio in rushing attack

Fig. 11. Packet delivery rate in adversarial scenario

Fig. 11(b) and Fig. 11(c) compares THMR 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 THMR. The packet delivery ratio in THMR keeps stable at above 97% even there are 10 attackers. The results show that THMR can resist black hole attack and rushing attack effectively. This is because: (1) THMR can exclude the unreliable routes in *Route Probe Phase* before transmitting packets; (2) THMR 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.

Fig. 12 depicts the normalized routing overhead. With the number of attackers increasing, the routing overhead in THMR also increases. When the number of attackers is 4, THMR incurs 33.6% more routing overhead than SRP. When the number of attackers is up to 10, THMR incurs 91.6% more routing overhead than SRP. However, THMR improves the packet delivery ratio from 52.3% of SRP to 97.9% when there are 10 attackers. THMR incurs more routing overhead than SRP as a result of two reasons. The first one is THMR needs to detect the unreliable routes before transmitting packets. The second one relates to the fact that in the presence of many attackers, the routes are more likely to become unreliable, thus THMR needs to invoke route discovery more often to find the fresher routes.

Fig. 13 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 THMR doesn't increase the end-to-end delay.

The routing convergent time in THMR depends on the timers (T_S and T_D) 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 0.3s, the convergent time in THMR is 0.4s, while it is 0.2s in SRP. THMR has longer convergent time because: (1) THMR tends to establish longer routes to prevent them from intersecting with each other; (2) THMR tends to find as many node-disjoint routes as possible in a route discovery attempt to prevent route discovery from being invoked frequently.

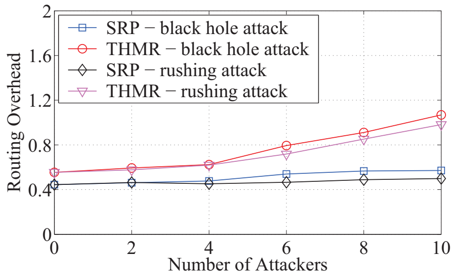


Fig. 12. Routing overhead in adversarial scenario

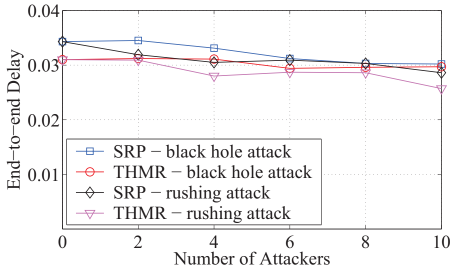


Fig. 13. End-to-end delay in adversarial scenario

VII. 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 for MANET. The paper also formally defines topology-hiding and proposes a Topology-Hiding Multipath Routing protocol (THMR). Performance evaluation shows that THMR has better capability of finding routes. THMR doesn't degrade the performance when there is no attack. While in the adversarial scenario, the simulation results show that THMR 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 THMR.

VIII. Acknowledgement

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

References

[1] C. K. Toh, A. N. Le, et al. Load balanced routing protocols for ad hoc mobile wireless networks. *IEEE Communications Magazine*, 47(8):78–84, 2009.

[2] M. K. Marina, and S. R. Das. Ad Hoc On-demand Multipath Distance Vector Routing. *Wiley Wireless Communications and Mobile Computing*, 6(7):969–988, 2006.

[3] L. Abusalah, A. Khokhar, et al. A Survey of Secure Mobile Ad Hoc Routing Protocols. *IEEE Communications Surveys and Tutorials*, 10(4):78–93, 2008.

[4] E. Gerhards-Padilla, N. Aschenbruck, et al. Detecting Black Hole Attacks in Tactical MANETs Using Topology Graphs. *IEEE Conference on Local Computer Networks (LCN)*, pages 1043–1052, 2007.

[5] F. N. Abdesselam, B. Bensaou, et al. Detecting and avoiding wormhole attacks in wireless ad hoc networks. *IEEE Communications Magazine*, 46(4):127–133, 2008.

[6] W. Galuba, P. Papadimitratos, et al. Castor: Scalable Secure Routing for Ad Hoc Networks. *IEEE conference on computer communications (InfoCom)*, 2010.

[7] Y. C. Hu, A. Perrig, et al. Rushing Attacks and Defense in Wireless Ad Hoc Routing Protocols. *ACM workshop on Wireless Security (WiSe)*, pages 30–40, 2003.

[8] J. R. Douceur. The Sybil Attack. *International Workshop on Peer-to-Peer Systems (IPTPS)*, pages 251–260, 2002.

[9] D. Johnson, Y. Hu, et al. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. *IETF RFC 4728*, 2007.

[10] S. Bengio, G. Brassard, et al. Secure implementation of identification systems. *Journal of Cryptology*, 4(3):175–184, 1991.

[11] P. Papadimitratos, and Z. J. Haas. Secure Routing for Mobile Ad Hoc Networks. *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, pages 27–31, 2002.

[12] P. Papadimitratos, and Z. J. Haas. Secure Data Communication in Mobile Ad Hoc Networks. *Journal on Selected Areas in Communications*, 24(2):343–356, 2006.

[13] M. Burmester, and B. Medeiros. On the Security of Route Discovery in MANETs. *IEEE Transaction on Mobile Computing*, 8(9):1180–1188, 2009.

[14] C. Perkins, E. Belding-Royer, et al. Ad hoc On-Demand Distance Vector (AODV) Routing. *IETF RFC 3561*, 2003.

[15] M. Rajabzadeh, F. Adibniya, et al. Adaptive DSR Protocol with Cooperative Agents for Different Mobility and Traffic Patterns. *International Conference on Systems and Networks Communications (ICSNC)*, pages 310–315, 2008.

[16] F. Rookhosh, A. T. Haghighat, et al. Disjoint Categories In Low Delay and On-demand Multipath Dynamic Source Routing Ad Hoc Networks. *International Conference on Distributed Framework and Applications (DFmA)*, pages 207–213, 2008.

[17] V. C. Frias, G. D. Delgado, et al. MM-DSR: Multipath QoS Routing For Multiple Multimedia Sources Over Ad Hoc Mobile Networks. *IEEE Latin America Transactions*, 5(6):448–456, 2007.

[18] L. F. Garcia, and J. M. Robert. Preventing Layer-3 Wormhole Attacks In Ad Hoc Networks With Multipath DSR. *IFIP Annual Ad Hoc Networking Workshop*, pages 15–20, 2009.

[19] E. K. Asl, and M. Damanafshan. EMP-DSR: An Enhanced Multi-path Dynamic Source Routing Algorithm for MANETs Based on Ant Colony Optimization. *International Conference on Modelling and Simulation*, pages 692–697, 2009.

[20] S. Adibi, and G. B. Agnew. Multilayer Flavoured Dynamic Source Routing In Mobile Ad Hoc Networks. *IET Communications*, 2(5), 2008.

[21] T. C. Huang, S. Y. Huang, et al. AODV-Based Backup Routing Scheme in Mobile Ad Hoc Networks. *International Conference on Communications and Mobile Computing (CMC)*, pages 254–258, 2010.

[22] Y. B. Yang, and H. B. Chen. An Improved AODV Routing Protocol for MANETs. *International Conference on Wireless Communications*, Networking and Mobile Computing (WiCom):1–4, 2009.

[23] X. Li, and L. Cuthbert. Stable Node-Disjoint Multipath Routing with Low Overhead in Mobile Ad Hoc Networks. *IEEE Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOT)*, pages 184–191, 2004.

[24] Z. Ye, S. V. Krishnamurthy, et al. A Routing Framework for Providing Robustness to Node Failures in Mobile Ad hoc Networks. *Elsevier Ad Hoc Networks*, 2(1):87–107, 2004.

[25] S. J. Lee, and M. Gerla. AODV-BR: Backup Routing in Ad hoc Networks. *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1311–1316, 2000.

[26] F. Kuhn, R. Wattenhofer, et al. Worst-Case Optimal and Average-Case Efficient Geometric Ad Hoc Routing. *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 267–278, 2003.

[27] V. Loscri, and S. Marano. A New Geographic Multipath Protocol for Ad hoc Networks to Reduce the Route Coupling Phenomenon. *IEEE Vehicular Technology Conference (VTC)*, pages 1102–1106, 2006.

[28] B. Kannhavong, H. Nakayama, et al. A Survey of Routing Attacks in Mobile Ad Hoc Networks. *IEEE Wireless Communications*, 14(5):85–91, 2007.

[29] LR.Reddy, and SV.Raghavan. SMORT: Scalable Multipath On-demand Routing for Mobile Ad Hoc Networks. *Elsevier Ad hoc Networks*, 5(2):162–188, 2007.