Preventing Cooperative Black Hole Attacks in Mobile Ad Hoc Networks: Simulation Implementation and Evaluation

Hesiri Weerasinghe and Huirong Fu, Member of IEEE
Department of Computer Science and Engineering
Oakland University
Rochester MI 48309 USA
{hdweeras, fu@oakland.edu}

Abstract

A black hole attack is a severe attack that can be easily employed against routing in mobile ad hoc networks. A black hole is a malicious node that falsely replies for any route requests without having active route to specified destination and drops all the receiving packets. If these malicious nodes work together as a group then the damage will be very serious. This type of attack is called cooperative black hole attack. In [9], we proposed a solution to identifying and preventing the cooperative black hole attack. Our solution discovers the secure route between source and destination by identifying and isolating cooperative black hole nodes. In this paper, via simulation, we evaluate the proposed solution and compare it with other existing solutions in terms of throughput, packet loss percentage, average end-to-end delay and route request overhead. The experiments show that (1) the AODV greatly suffers from cooperative black holes in terms of throughput and packet losses, and (2) our solution proposed in [9] presents good performance in terms of better throughput rate and minimum packet loss percentage over other solutions, and (3) our solution proposed in [9] can accurately prevent the cooperative black hole attacks.

1. Introduction

A mobile ad hoc network (MANET) is a collection of wireless mobile nodes which have the ability to communicate with each other without having fixed network infrastructure or any central base station. Since mobile nodes are not controlled by any other controlling entity, they have unrestricted mobility and connectivity to others. Routing and network management are done cooperatively by each other nodes. Due to limited transmission power, multi hop architecture is needed for one node to communicate with another through network. In this multi hop architecture, each node works as a host and as well as a router that forwards packets for other nodes that may not be within a direct communication range. Each node participates in an ad hoc route discovery protocol which finds out multi hop routes through the mobile network between any two nodes. These infrastructure-less mobile nodes in ad hoc networks dynamically create routes among themselves to form own wireless network on the fly. Thus, mobile ad hoc networks provide an extremely flexible communication method for any place where geographical or terrestrial constraints are present and need network system without any fixed architecture, such as battlefields, and some disaster management situations.
Recent research on MANET shows that the MANET has larger security issues than conventional networks [1,2]. Any security solutions for static networks would not be suitable for MANET. Zhou et al. [1] and Lundberg [3] discussed several types of attacks that can easily be performed against a MANET. In the black hole attack, malicious nodes provide false routing information to the source node whose packets they want to intercept. In denial of service attacks, malicious node floods the targeted node so that the network or the node no longer operates correctly. In route table overflow attacks, an attacker tries to create lots of routes to non-existence nodes and overflows the routing tables. In impersonation attacks, malicious node may impersonate another node while sending the control packets to create an anomaly update in routing table. In this paper, we will focus on the black hole and cooperative black hole attacks.

The main contributions of this work are threefold. First, we implement the simulation of the solutions proposed for the cooperative black hole attacks by Ramaswamy et al. [9]. Second, we also add some changes to the algorithm to improve the accuracy in preventing black hole attacks. For example, previously the algorithm does not check current intermediate node for black hole if the next hop is not reliable. This proposed algorithm does not give any details about the implementation of the algorithm. In this paper we completely describe the implementation details which we address the several issues which are not considered in [9]. Finally, we compare the performance of the modified solution with other existing solutions in terms of throughput, end-to-end delay route request overhead, and packet lost percentage.

The rest of the paper is organized as follows. In section 2, we introduce the related works. Next, in section 3, we present the implementation details. Then in section 4, we describe the details about performance evaluation metrics and in section 5 we describe the simulation setup and scenarios, and analyze and compare the solutions in details. Then in section 6, we analyze the security of the new algorithm. Finally, we conclude in section 7.

2. Related works

The routing protocols proposed for MANETs can be classified into four broad categories [11]: Flat routing, Hierarchical routing, GPS routing, and Power based routing. Flat routing is the most widely used category. These flat routing protocols can be further classified into two main subgroups [12]: table driven and on-demand routing protocols. The table driven routing protocol is a proactive scheme in which each node maintains consistent and up to date routing information to every other node in the network. Every routing change in the network should be propagated through the network in order to maintain consistent routing information. In the on-demand routing (reactive routing), any node creates route only when it needs to send some data to the destination. The source node initiates route discovery process when necessary.

There are three main routing protocols proposed for MANETs [4]: Ad hoc On demand Distance Vector (AODV) [5] routing, Dynamic Source Routing (DSR) [6], and Destination Sequence Distance Vector routing (DSDV) [7]. AODV and DSR belong to on-demand routing protocols and DSDV is a table-driven routing protocol. In this paper, we focus on AODV. However, the proposed solution is also applicable to other on-demand protocols, such as DSR.

The AODV protocol is vulnerable to the well-known black hole attack. A black hole is a node that always responds positively with a RREP message to every RREQ, even through it
does not really have a valid route to the destination node. Since a black hole node does not have to check its routing table, it is the first to respond to the RREQ in most cases. Then the source routes data through the black hole node, which will drop all the data packets it received rather than forwarding them to the destination. In this way the malicious node can easily misroute lot of network traffic to itself and could cause an attack to the network with very little effort on it. These black hole nodes may work as a group. That means more than one black hole nodes work cooperatively to mislead other nodes. This type of attack is called cooperative black hole attack.

Researchers have proposed solutions to identify and eliminate black hole nodes [8-10]. In [8], Deng et al. proposed a solution for individual black holes. But they have not considered the cooperative black hole attacks. According to their solution, information about the next hop to destination should be included in the RREP packet when any intermediate node replies for RREQ. Then the source node sends a further request (FREQ) to next hop of replied node and asks about the replied node and route to the destination. By using this method we can identify trustworthiness of the replied node only if the next hop is trusted. However, this solution can not prevent cooperative black hole attacks on MANETs. For example, if the next hop also cooperates with the replied node, the reply for the FREQ will be simply “yes” for both questions. Then the source will trust on next hop and send data through the replied node which is a black hole node.

Ramaswamy et al. [9] proposed a solution to defending against the cooperative black hole attacks. But in [9], no simulations or performance evaluations have been done. Ramaswamy et al. [13] studied multiple black hole attacks on mobile ad hoc networks. However, they only considered multiple black holes, in which there is no collaboration between these black hole nodes. In this paper, we evaluate the performance of the proposed scheme in defending against the collaborative black hole attack. In [10], Yin et al. proposed a solution to defending against black hole attacks in wireless sensor networks. The scenario that they considered in sensor networks is quite different than MANETs. They consider the static sensor network with manually deployed cluster heads. They did not consider the mobility of nodes. Also they have one sink node and all sensors send all the data to the sink. Each node needs to find out the route only to the sink. Since this scenario is not compatible with MANET, we are not going to discuss it further.

In this paper we simulate the algorithm proposed by [9] with several changes to improve the accuracy of preventing cooperative black hole attacks and to improve the efficiency of the process. We also simulate AODV and the solution proposed by [8] and compares them with [9].

3. Protocol implementation

Our algorithm uses a methodology to identify multiple black hole nodes working collaboratively as a group to initiate cooperative black hole attacks. This protocol is a slightly modified version of AODV protocol by introducing Data Routing Information (DRI) table and cross checking using Further Request (FREQ) and Further Reply (FREP).

3.1. Data routing information table

Each node maintains a data routing information (DRI) table. This table keeps track of whether or not the node did data transfers with its neighbors. This table contains one entry for each neighbor and indicates whether the node has sent data through this neighbor and whether
the node has received data from this neighbor. Table entry contains node id, from and through as shown in Table 1. The field from stands for information on routing data packets from the node (in the node id field) whiles the field through stands for information on routing data packets through the node (in the node id field). Values of from and through fields will be 0 or 1 to represent false and true respectively. Table 1 shows the sample DRI table for a node. The entry 1,0 for node 3 implies that this node has routed data packets from node 3 but has not routed any data packet through node 3. The entry 1,1 for node 6 implies that this node has successfully routed data packets from and through node 6. The entry 0,0 for node 2 implies that node has not routed any data packets from or through node 2.

This DRI table is updated when any node received data packet from one of its neighbors or any node that sent data packets through one of its neighbors. In addition, if any node finds out the reliable path to destination which it needs to send the data, DRI table is updated with entries for all intermediate nodes through the path. This reliable route discovery process will be described in details in the following section III .B.

<table>
<thead>
<tr>
<th>Node id</th>
<th>Data Routing Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

3.2. Algorithm

In our new technique, we rely on reliable nodes (any node is reliable to the source node if the source has successfully routed data through that node) to transfer data packets. The modified AODV routing protocol and the algorithm for our proposed methodology are described below.

In this protocol, if the source node (SN) does not have the route entry to the destination, it will broadcast a RREQ (Route Request) message to discover a secure route to the destination node same as in the AODV. Any node received this RREQ either replies for the request or again broadcasts it to the network depending on the availability of fresh route to the destination. If the destination replies, all intermediate nodes update or insert routing entry for that destination since we always trust destination. Source node also trusts on destination node and will start to send data along the path that reply comes back. Also source node will update the DRI table with all intermediate nodes between source and the destination.

If the intermediate node (IN) generates the Route Reply (RREP), it has to provide its next hop node (NHN) and its DRI entry for the next hop node. When the reply comes back, it collects the IP addresses of all nodes between source and the intermediate node but no intermediate node updates the route entry for the destination. Upon receiving RREP message from IN, the source node will check its own DRI table to see whether IN is a reliable node or not. If the source node has used IN before to route data, then IN is a reliable node and source will first send a route establishment message to IN node along the path that RREP comes according to the information contains in the RREP message. Upon receiving this message all nodes between the source and the intermediate node will update or insert route entry for the destination. Then source node starts sending data through the IN and updates the DRI table with nodes between source and IN node.
If the source has not routed data through IN before, IN is not a reliable node. Then source first stores the information about IN and the nodes between the source and IN, and sends Further Request (FREQ) message to NHN of the IN to verify the reliability of the IN and ask NHN:

1) Whether the IN has routed data packet through NHN.
2) Who is the current NHN’s next hop to the destination?
3) Has the current NHN routed data through its own next hop?

Then NHN in turn responds with Further Reply (FREP) message which includes:

1) DRI entry for IN.
2) The next hop node (NHN) of current NHN, and
3) The DRI entry for the current NHN’s next hop.

If the current NHN is the destination, then the next hop entry and the DRI entry for the next hop fields of FREP contain zeros and all intermediate nodes will either update or insert route entry for the destination. When the source receives FREP from destination, it starts routing data and updates its DRI table with all nodes between the source and the destination. If NHN is not the destination, based on the FREP message from NHN, the source node checks whether NHN is a reliable node or not. If the source node has routed data through NHN before, NHN is reliable; otherwise NHN is unreliable. Also the source node will check whether IN is a black hole or not. If the second bit of the DRI entry from IN is equal to 1 (i.e. IN has routed data through NHN) and the first bit of the DRI from NHN for IN is equal to 0 (i.e. NHN has not routed data from IN) then IN is a black hole node. Also, if the current NHN’s next hop is an already visited node (node between current NHN and the IN that reply for the RREQ) then current NHN is a black hole node. If the current IN or NHN is a black hole node then source node identifies all the nodes in the reverse path from current IN or NHN to the node that generate RREP as black hole nodes. Then source node starts the secure route discovery process from beginning and sends the RREQ again. Source node ignores any other RREP messages from any black hole nodes and broadcasts the list of cooperative black holes to notify others.

If IN is not a black hole node and the NHN is a reliable node, then route to destination is secure. Source node will update its DRI table with entries for all nodes from source to IN with 01 and start routing data via IN. If the NHN is an unreliable node, the source node treats the current NHN as IN and send FREQ to the updated IN’s next hop node and goes into the steps described above.

4. Performance metrics

To evaluate the performance of our solution and compare with AODV and the solution proposed by Deng et al. [8], we consider several performance metrics. The existence of black hole nodes in a mobile ad hoc network directly causes to packet loss in between the source and the destination. This will also affect the application throughput between source and destination. We select the application throughput ratio as one performance metric and data packet loss as another metric. We also consider the end to end delay of application data as another performance metric. Since this protocol uses more control packets, we need to find out the control packet overhead that the solution introduced. Then we select control packet overhead as another performance evaluation metric. Next we describe the above four metrics in details.
4.1. Throughput ratio

The throughput is the number of bytes transmitted or received per second. The throughput ratio, denoted by $T$, is calculated as follows:

$$T = \frac{\sum_{i=1}^{n} T_i^r}{\sum_{i=1}^{n} T_i^s} \times 100\%$$

where $T_i^r$ is the average receiving throughput for the $i^{th}$ application, $T_i^s$ is the average sending throughput for the $i^{th}$ application, and $n$ is the number of applications.

4.2. Packet loss percentage

Data packet loss rate, $L$ is calculated as follows:

$$L = \frac{\sum_{i=1}^{n} (N_i^s - N_i^r)}{\sum_{i=1}^{n} N_i^s} \times 100\%$$

where $N_i^s$ and $N_i^r$ are the number of application data packets sent by the sender and the number of application data packets received by the receiver, respectively for the $i^{th}$ application, and $n$ is the number of applications.

4.3. Average end-to-end delay

Average end-to-end delay of the application data packets, denoted by $D$, is calculated as follows:

$$D = \frac{\sum_{i=1}^{n} d_i}{n}$$

where $d_i$ is the average end-to-end delay of data packets of $i^{th}$ application and $n$ is the number of CBR applications.

4.4. Control packet overhead

Control packet overhead is the ratio of the number of control data bytes which is used by the sender to discover the secure route between sender and receiver and the total number of application data bytes transferred between sender and receiver. This is denoted by $O$ and calculated as follows:

$$O = \frac{\left(\sum_{i=1}^{n} rqq_i + \sum_{i=1}^{n} rrt_i\right) \times \text{cpsize}}{\sum_{i=1}^{n} (N_i^s \times \text{dpsize}) + \left(\sum_{i=1}^{n} rqq_i + \sum_{i=1}^{n} rrt_i\right) \times \text{cpsize}} \times 100\%$$

where $rqq_i$ is the number of route requests sent by the sender and $rrt_i$ is the number of route request retries done by the sender. $\text{cpsize}$ is the size of the request packet in bytes and $\text{dpsize}$, is the size of the application data packet in bytes.
is the size of the application data packet in the \( i \)\textsuperscript{th} application. \( N_i \) is the number of data packets sent by the \( i \)\textsuperscript{th} application source and \( n \) is the number of applications.

5. Simulation and results

In this section we present a set of simulation experiments to evaluate this protocol by comparing with the original AODV [5] and the solution proposed by Deng et al. at the University of Cincinnati [8].

5.1. Experimental settings

To evaluate and compare the performance of our protocol, we implement our protocol and the protocol proposed by Deng et al. at University of Cincinnati [8]. All the variable parameters used in the simulations are default values which are recommended by the AODV [5]. To measure the performance evaluation metrics which we described in the previous section, we consider several scenarios. We select the number of black hole nodes, total number of nodes, mobility speed of the nodes and terrain area as variable parameters and we simulate all protocols for different settings and collect the values of performance evaluation metrics in each scenario. Unless explicitly says, all simulation scenarios are configured according to the Table 2. Wherever Times New Roman is specified, Times Roman, or Times may be used. If neither is available on your word processor, please use the font closest in appearance to Times New Roman that you have access to. Please avoid using bit-mapped fonts if possible. True-Type 1 fonts are preferred.

<table>
<thead>
<tr>
<th>TABLE 2: SCENARIO SPECIFICATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of nodes</td>
</tr>
<tr>
<td>Number of cooperative black hole nodes</td>
</tr>
<tr>
<td>Average mobility speed</td>
</tr>
<tr>
<td>Terrain area</td>
</tr>
</tbody>
</table>

We use 5 Constant Bit Rate (CBR) sessions in the mobile ad hoc network. Each of these CBR applications uses 512-byte data packets at the rate of 2 packets/second. The random waypoint model is used to model mobility. Each node starts its journey from a random location to a random destination. When the destination is reached, another random destination is targeted after 30 second pause. We vary the mobile speed according to the scenarios. We conduct 100 independent simulation runs for each scenario with 1-100 seed values to obtain the average measures for the performance metrics. We simulate 250s for each run. To compare the performance of protocols, we add a benchmark scenario for figures 2 to 4. This scenario contains AODV without having any black hole nodes. This plot is shown in dotted lines in Figures 2-4.

5.2. Impact of number of black hole nodes

We first studied the performance of our protocol with varying number of black hole nodes.
Figure 1 shows the impact of the number of black hole nodes in the network on all four performance metrics discussed above. Moreover, in each graph, the number of black hole nodes varies from 2 to 10 with the increment of 2 and all other configurations are specified as table 2.

Figure 1(a) illustrates the impact of the number of black hole nodes on throughput for all protocols. First, as this figure depicts, AODV and the solution proposed by [8] heavily suffer from the cooperative black hole attacks since both protocols do not have any mechanism to prevent cooperative black hole attacks. Moreover, the throughput of AODV goes down under 20% regardless of the number of black hole nodes in the network. Second, our protocol gave higher throughput than the other two protocols since our protocol prevents malicious packet drops by black holes. Even with the 10 black hole nodes, our protocol produces more than 65% throughput. Furthermore, all protocols present lesser throughput when the number of black holes is higher since there are more attacks.

Figure 1(b) shows the impact of the number of black hole nodes on end-to-end delay. First, our protocol has little bit of more end-to-end delay than others since it takes more time to find out a secure route. Therefore, this will be the tradeoff between delay and the packet loss. Second, when the number of cooperative black holes increases, our solution increases the delay since it has to avoid more malicious nodes when it tries to find out secure route from source to destination. Furthermore, the solution proposed by [8] does not have much difference when the number of black holes increases since it always checks only one next hop node. Interestingly, since more black holes reply quickly for the route requests, the AODV decreases the delay when the number of black hole nodes increases.

Figure 1(c) demonstrates the impact of the number of black hole nodes on route request overhead. First, the AODV introduces the lowest overhead which is less than 2% since it does not use any further requests for finding secure routes. It also decreases the route request overhead when the number of black holes increases since it receives more responses from black hole nodes for every route request initiated and uses less route requests. Second, the solution proposed by [8] also introduced less amount of overhead with the cooperative black holes since it checks only the first next hop of intermediate node. Since the existence of more cooperative black hole nodes makes our protocol to use more and more further requests to identify and eliminate all black hole nodes, our protocol introduces little more overhead when number of black holes increases. Thus, there is a tradeoff between the route request overhead and the packet lost rate when cooperative black holes exist.

Figure 1(d) illustrates the impact of the number of black hole nodes on packet loss. First, it properly depicts that AODV and the solution proposed by [8] heavily suffered by the cooperative black hole attacks. Therefore, its packet loss percentage goes higher 70% regardless of the number of black hole nodes in the network. Second, the packet loss of the solution proposed by [8] is very high but lower than AODV since AODV does not prevent any type of black hole attacks while the other solution prevent only individual black hole attacks but not cooperative black hole attacks. Third, our protocol gave much lower packet loss percentage than two other protocols since ours prevents both individual and cooperative black hole attacks. When number of black hole nodes is increased, all the protocols increase packet loss percentage. Furthermore, packet loss of our protocol increase slowly with the number of black hole nodes since higher route finding delay caused for little more packet losses. However, the AODV and the other solution significantly increase their packet loss percentage with the number of black hole nodes since the frequency and the capacity of attacks increases which can not be prevented by these two protocols.
5.3. Impact of number of nodes

We studied the performance of our protocol with varying number of nodes. Figure 2 shows the impact of the number of nodes on the protocols. In each graph, the number of nodes is varying from 20 to 90 with the step of 10 and all other configurations are specified as table 2.

Figure 2(a) illustrates the impact of the number of nodes on throughput for all protocols. We make several observations on Figure 2(a). First, the AODV and the solution proposed by [8] heavily suffer from the cooperative black hole attacks since both protocols do not have any mechanism to defend against cooperative black hole attacks. Thus, the throughput goes down under 50% regardless of the number of nodes in the network for AODV and the solution proposed by [8]. Second, our protocol presents higher throughput nearly as high as the AODV without black holes (benchmark plot in the figure) since our protocol can prevent malicious packet drops by black holes. But the throughput of the new protocol is slightly lower than the benchmark regardless of the number of nodes. The major reason is that our protocol takes more time to avoid cooperative black holes to establish a secure route. Therefore, this delay introduces packets loss. Third, when the number of nodes is 20 all the protocols gave somewhat lower throughput since 20 nodes are not quite enough to cover the 1500 x 1500 m² region for successful communication. Last, when the number of nodes increases, the throughput increases for all protocols since source nodes can find more and more routes to destinations which can avoid black hole nodes (some normal nodes can reply before the black hole nodes).

Figure 2(b) shows the impact of the number of nodes on the end-to-end delay. First, the end-to-end delay does not depict much difference when the number of nodes increases because the source and destination pairs of all CBR applications are the same for each scenario regardless of the number of nodes. But for less number of nodes, all protocols take more time since alternative routes are limited. Second, our protocol has little bit of more end-to-end delay than others since it takes more time to find out a secure route. Thus, this is the tradeoff between delay and the packet loss.

Figure 2(c) demonstrates the impact of the number of nodes on route request overhead. First, the AODV introduces 3-4% route request overhead when there are no black hole nodes since it does not use any further requests. The solution proposed by [8] also gives nearly a same amount of overhead with the cooperative black holes. Moreover, since it checks only the first next hop of intermediate node, it does not use further requests to find out a secure route. Second, our protocol uses more route requests and further requests since it checks each and every next hop until it finds a reliable next hop. Thus, our protocol gives 6-8% more route request overhead than the other two protocols. There is a tradeoff between route request overhead and the packet lost rate when cooperative black holes exist. Third, for all the route request overhead than the other two protocols. There is a tradeoff between route request overhead and the packet lost rate when cooperative black holes exist. Third, for all the protocols, the overhead goes down when the number of nodes increases. In our protocol, the route request overhead goes down significantly with the increase of the number of nodes since the existence of more nodes gives more alternative routes for further request process which will reduce the number of request and retries needed. The AODV makes the lowest overhead when black holes are present since AODV uses less route requests when black holes reply quickly for route requests.
Figure 1: Impact of the number of black hole nodes

(a) Throughput vs number of black hole nodes

(b) End-to-end delay vs number of black hole nodes

(c) Route request overhead vs number of black hole nodes

(d) Packet loss vs number of black hole nodes

Figure 2: Impact of the number of nodes

(a) Throughput vs number of nodes

(b) End-to-end delay vs number of nodes

(c) Route request overhead vs number of nodes

(d) Packet loss vs number of nodes
Figure 2(d) illustrates the impact of the number of nodes on packet loss. First, it properly
depicts that the AODV and the solution proposed by [8] heavily suffer from the cooperative
black hole attacks. For example, the packet loss percentage goes higher than 60% regardless
of the number of nodes in the network. Therefore, our protocol presents much lower packet
loss percentage than the solution proposed by [8]. Second, when the number of nodes is 20 all
the protocols present somewhat higher packet loss percentage since 20 nodes are not quite
enough to cover the 1500 x 1500 m² region for successful communication. When the number
of nodes increases, the packet loss percentage decreases for all the protocols since sources
would find more routes to destinations, which can avoid black hole nodes because some non-
black hole nodes can reply before the black hole nodes.

5.4. Impact of mobility speed

We study the performance of our protocol with varying mobility speed of nodes. Figure 3
shows the impact of mobility speed of nodes on all four performance metrics discussed
above. In each graph, the mobility speed varies from 5m/s (11.2 miles/hour) to 75m/s (168
miles/hour) and all other configurations are specified as table 2. This speed range covers large
varieties of mobile ad hoc networks.

Figure 3(a) illustrates the impact of the mobility speed on throughput. It depicts that the
AODV and the solution proposed by [8] heavily suffer from the cooperative black hole
attacks. For example, the throughput goes down under 50% regardless of the mobility speed.
Our protocol gave higher throughput nearly as high as the benchmark since our solution
prevents cooperative black hole attacks properly. For example, when the mobility speed is
5m/s, our protocol presents the highest throughput. Second, the throughput of our protocol
slightly decreases when the mobility speed increases since high mobility speed causes higher
link breakdown probability, and in turn the protocol introduces more route discovery
processes. Thus, our protocol spends more time to find secure routes which causes slightly
lower throughput. Third, the throughput of the solution proposed by [8] and AODV, however,
slightly increases when the speed increases. This can be explained as: more link breakdowns
will cause more new route discovery processes and in turn the number of packets that black
holes drop may slightly reduce since new routes can be found to avoid black holes.

Figure 3(b) demonstrates the impact of mobility speed of nodes on end-to-end delay. The
end-to-end delay increases when the mobility speed increases because more link breakdowns
increase. Our protocol has a larger end-to-end delay than others since it takes more time to
find out a secure route. Thus, this will be the tradeoff between delay and the packet loss.

Figure 3(c) illustrates the impact of the mobility speed of nodes on route request overhead.
First, AODV gives 2-4 % route request overhead when there are no black hole nodes.
Furthermore, the solution proposed by [8] gives slightly higher amount of 3-5% overhead
with the cooperative black holes since they check only the first next hop of intermediate
nodes. Since our protocol uses more route request and further requests to check every next
hop, our protocol presents 4-5% more route request overhead than the other two protocols.
However, in all the protocols, the route request overhead goes higher with the mobility speed
of nodes. Since higher speed of nodes make more link breakdowns, the number of route
discoveries will increase. Thus, it will increase the number of request and retries that
introduce more overhead.

Figure 3(d) illustrates the impact of the mobility speed of nodes on packet loss percentage.
We make several observations on the figure. First, it depicts that the AODV and the solution
proposed by [8] heavily suffer from the cooperative black hole attacks. For example, the packet loss percentage goes as high as 50% percent regardless of the mobility speed. However, our protocol presents much lower packet loss percentage than the solution by [8]. For example, when the mobility speed is 5m/s our protocol gives the lowest packet loss percentage and it increases with the mobility speed since low mobility speed causes lower link breakdown probability and less number of secure route discovery processes. When the mobility speed increases, the packet loss rate of our new protocol goes up since more link break downs make our protocol spend more time to find secure routes which makes higher packet loss. Second, the packet loss percentage of our new protocol is higher than the benchmark because our protocol takes more time to avoid cooperative black hole and establish a secure route when the number of link breakdown increases. Thus, this delay causes higher packet loss percentage than the benchmark. Third, packet loss percentage of the AODV and the solution proposed by [8] slightly decreases when the speed increases since more link breakdowns will cause more new route discoveries. Therefore, the number of packets that black holes drop may slightly reduce since new routes can be found to avoid black holes.

5.5. Impact of terrain area

We also study the performance of our protocol with varying the size of terrain area. Terrain area varies from 500x500 m² to 750x750 m², 1000x1000 m² to 1250x1250 m², and to 1500 x 1500 m². All other configurations are specified as table 2.

Figure 4 shows the impact of terrain area on the protocols. We make several observations on Figure 4(a). First, the AODV and the solution proposed by [8] heavily suffer from the cooperative black hole attack. For example, the throughput goes down under 40% regardless of the terrain area for the AODV and the solution proposed by [8]. However, our protocol presents that the throughput is nearly as high as the AODV without black holes. Second, the throughput decreases with the increase of terrain area since when the distance between the source and destination becomes smaller, it is easier to find secure routes in small terrain area than in large terrain area. Third, the throughput of the solution proposed by [8] and AODV slightly increases when the terrain area increases since the distance between black holes and sources may slightly increases with the large area. Therefore, the number of packets that black holes drop may slightly reduce since alternative routes can be found to avoid some black holes.

Figure 4(b) illustrates the impact of terrain area on end-to-end delay. The end-to-end delay increases when the terrain area increases because the distance between source and the destination increases. However, our protocol has higher end-to-end delay than others since it takes more time to find out a secure route. Thus, this is the tradeoff between delay and the packet loss.

Figure 4(c) demonstrates the impact of terrain area on route request overhead. First, the AODV with no black hole nodes and the solution proposed by [8] with the cooperative black holes introduce nearly the same amount of overhead. Since they check only the first next hop of intermediate node they use less route requests to find out a secure route. Since our protocol uses more route requests and further requests, our protocol presents more route request number of route discoveries increases with the increase of distance. Therefore, it will increase the number of requests and retries, which introduces more overhead. Moreover, the overhead of our protocol increases quickly with the terrain area since more route discoveries for secure
route increase the number of requests needed than the other protocols.

Figure 3: Impact of the mobility speed

(a) Throughput vs mobility speed
(b) End-to-end delay vs mobility speed
(c) Route request overhead vs mobility speed
(d) Packet loss vs mobility speed

Figure 4: Impact of the terrain area

(a) Throughput vs terrain area
(b) End-to-end delay vs terrain area
(c) Route request overhead vs terrain area
(d) Packet loss vs terrain area
Figure 4(d) presents the impact of terrain area on packet loss percentage. It depicts that the AODV and solution proposed by [8] heavily suffer from the cooperative black hole attacks. For example, its packet loss percentage goes as high as 70% without regarding the terrain area. Moreover, when the terrain area is 500x500m², our protocol presents the lowest packet loss percentage, which increases with the increase of terrain area. Since increase in distance between source and destination increases the number of secure route discovery processes AODV and our protocol introduces higher packet loss when the terrain area increases. Moreover, the packet loss percentage of our new protocol is higher than the AODV without black hole because our protocol takes more time to avoid cooperative black hole and establish a secure route than the AODV. Therefore, this delay causes higher packet loss percentage than the AODV without black holes. However, our protocol presents low packet loss, almost the same as the benchmark since it prevents cooperative black hole attacks. Furthermore, the packet loss percentage of the solution proposed by [8] slightly decreases when the terrain area increases since the distance between black holes and sources may slightly increase with the larger area.

6. Security analysis

The main goal of this protocol is to detect and prevent cooperative black hole attacks. In this section we analyze the security of the proposed protocol.

Since source node only trusts reliable nodes, all other intermediate nodes will be explicitly checked for black hole nodes. If any group of malicious nodes work together to compromise the network, they will collaborate to provide false information to the controlling protocols to mislead any attack prevention mechanism. Since AODV depends only on one intermediate node, one malicious node can compromise the whole network and may partition the whole network. The protocol proposed by [8] depends only on two intermediate nodes. Thus, compromising two nodes may be able to compromise the whole network. But with the new protocol, network can not be compromised with any number of individual or collaborative black hole nodes since source node does not forward data until it finds the secure and reliable route to the destination. Large number of collaborative black hole nodes can affect the network by increasing the end to end delay only for the first route discovery. Then, the protocol finds out all collaborative black hole nodes which involves in the attack and disseminates them in the network. After that, no node accepts replies from those black hole nodes. Generally, we assume the percentage of black hole nodes in the network is quite less than the percentage of legitimate nodes.

Since any intermediate node does not update route entry for the destination until the source node confirms the secure route, the risk of adding false route entries to the routing table of the intermediate node is removed. This protocol also prevents route discovery loops. If a malicious node refers an already visited node as its next hop, then the secure route discovery process will be in endless loop. Since the source node temporarily keeps track of all visited nodes, this new algorithm can prevent such infinite loops.

7. Conclusion

In this paper, we studied the problem of cooperative black hole attacks in MANET routing. We simulated our proposed solution [9] and the currently available solution proposed by Deng et al. [8] using the QualNet simulator and compared the performance of [8,9] with the original AODV in terms of throughput, packet loss rate, end-to-end delay and control packet...
overhead. Simulation results show that (1) the AODV and the solution proposed by Deng et al. [8] greatly suffer from cooperative black holes in terms of throughput and packet losses. (2) our solution presents good performance in terms of better throughput rate and minimum packet loss percentage over other solutions.

8. Acknowledgement

This material is based upon work in part supported by the National Science Foundation (NSF) under Grant No. 0716527, Michigan Space Grant Consortium (MSGC) and Faculty Research Fellowship at Oakland University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF, MSGC or Oakland University. The authors may be reached by email: fu@oakland.edu, phone: +1 (248) 370-4456, or fax: +1 (248) 370-4625.

9. References

V. Karpijoki, “Security in Ad Hoc Networks,” http://www.hut.fi/~vkarpijo/netsec00/netsec00_manet_sec.ps
Authors

Hesiri Weerasinghe is currently a Ph.D student in Computer Science and Engineering Department, Oakland University, Michigan, USA. He received his B.Sc degree in Mathematics from University of Kelaniya of Sri Lanka in 2000 and his M.Sc degree in Computer Science and Engineering from Oakland University in 2006. His research interests include Information and Network Security, Mobile ad-hoc networks and Communication Networks.

Dr. Huirong Fu joined Oakland University as an assistant professor in 2005. Previously, she has been working as an assistant professor at North Dakota State University (NDSU) for three years, and as a post-doctoral research associate at Rice University for more than two years. As a lead professor and the principal investigator in several projects funded by the NSF, Dr. Fu has been actively conducting research in the area of information security. Her primary research interests are in information assurance and security, networks, Internet data centers, and multimedia system and services.