Detecting Black-hole Attack in Mobile Ad Hoc Networks

Bo Sun Yong Guan Jian Chen Udo W. Pooch
Department of Computer Science
Texas A&M University
College Station, TX 77843-3112

Abstract—This paper presents a general approach for detecting the black-hole attack in mobile ad hoc networks, which are particularly vulnerable to attacks compared to traditional wired networks due to its mobility and broadcast in nature. In particular, black-hole attacks can be easily deployed by the adversary. To defend against this attack, we devise a neighborhood-based method to detect whether there exists a black hole attack and a routing recovery protocol to set up a correct path to the true destination. Our methods have the remarkable advantage that the number of encryption/decryption operations for authentication is much reduced compared to those methods completely relying on cryptography-based authentication, which can save many system resources (e.g., reduce the energy consumption). Through simulation, we evaluate these methods in terms of packet throughput, routing control overhead, detection probability, false positive probability and false negative probability. Simulation results show that our methods can effectively detect black hole attacks in the sense that detection probability in most cases (the probability that one attacker might be detected) is above 93%. Meanwhile, it does not introduce much routing control overhead. With the routing recovery mechanism, the packet throughput can be improved by at least 15% and the false positive probability of the detection approach is very low (less than 1.7%).

Keywords—Network Security, Mobile Ad Hoc Network, Routing

1. Introduction

Mobile Ad-hoc NETworks (MANETs) have seen tremendous growth in recent years. It is a new paradigm of communication in which there is no fixed infrastructure. Nodes within the radio range of each other can communicate directly over the wireless link, while those that are far apart use other nodes as relays.

Routing protocols are the cornerstone of MANET. In the past few years, much research efforts have been focused on this area and many different kinds of routing protocols have been put forward in the literature, such as Wireless Routing Protocol (WRP) [1], Dynamic Source Routing protocol (DSR) [2], Ad hoc On Demand Distance Vector protocol (AODV) [3] and Location Aided Routing [4]. However, from the beginning of its design, almost none of the routing protocols specify security measures, but the nature of wireless ad hoc networks makes them very vulnerable to malicious attacks compared to traditional wired networks.

An attack occurs when an intruder tries to exploit vulnerabilities of a system. There are many types of attacks in MANET. Generally speaking, these attacks can be classified into two broad categories: passive and active attacks [6][9]. In passive attacks, the attackers typically involve eavesdropping of data, thus disclose the information of the location and move patterns of mobile nodes. This kind of attack is very difficult to detect, because the attacker seldom exhibits abnormal activities. Active attacks, on the other hand, involve actions performed by intruders. The target of the attack can be either data traffic or routing traffic [6]. The intruders may insert large volume of extraneous data packets into networks. They can also intentionally drop, corrupt and delay data packets passing through it.

In this paper, we are focusing on detecting a special active attack - black hole attack. One type of black hole attack can occur when the malicious node on the path directly attacks the data traffic by intentionally dropping, delaying or altering the data traffic passing through it. This type of black hole attack can be easily mitigated by setting the promiscuous mode of each node and listening to see if the next node on the path forward the data traffic as expected. Another type of black hole attack is to attack routing control traffic. The malicious node can impersonate some other node and advertise itself having the shortest path to the data source whose packets it is interested in. In this way, this malicious node becomes a black hole since the data traffic are misrouted to a wrong destination. We develop methods to detect this type of routing misbehavior caused black hole attack.

To defend against this type of black hole attack, we propose a neighborhood-based method. Our solution can be briefly elaborated as: Once the normal path discovery procedure in a routing protocol is finished, the source node sends a special control packet to request the destination to send its current neighbor set. By comparing the received neighbor sets, the source node can determine whether there is a black hole attack in the network. To mitigate the impact of the black hole attack, we design a routing recovery protocol to establish the path to the correct destination.

Our design has been motivated by three main factors: 1) Cryptography-based methods are expensive in terms that they require a priori trust and excessive overhead and resource consumption caused by the encryption/decryption operations for authentication purposes. The solution should have acceptable detection probability and improve the packet throughput to a reasonable level without having too much overhead and expensive resource requirements. This is important to MANET because mobile nodes are typically quite limited in their capacities: processing speed, memory space, link bandwidth and battery power. 2) The solution should be scalable and decentralized to operate in a large-scale network. And 3) The solution can be easily deployed without or with least modification to the existing routing protocol.

Our methods proposed in this paper have such remarkable advantage that the number of cryptography operations is much reduced compared to those that completely rely on cryptography-based methods. We have done a set of simulations using ns-2 [8] to evaluate the effectiveness and efficiencies of our methods. Our simulation result shows that the detection probability in most cases is above 93% without introducing too much overhead. The response mechanism can improve the packet throughput by at least 15%. The false positive probability is also very low, usually less than 1.7%.
II. Threat Model

In this section, we use AODV routing algorithm as the exemplary routing protocol to illustrate in detail the black hole attack.

A black hole attack happens when a malicious node D' intercepts the data traffic from the source node S to the destination node D. D' may misbehave by agreeing to forward packets but fail to do so, because it is overloaded, selfish, malicious, or broken [7]. This kind of black hole attack can be detected by setting the promiscuous mode of each node proposed in [7]. Another type of black hole attack is that D' claims to have the IP address of D, thus leads S to form the path to D', instead of D. Taking AODV for example, when the attacker D' receives a route request (RREQ) packet, it generates a route reply (RREP) packet to reply back to S telling S that it is the destination node. When D' is closer to the source node than the true destination D, a forged route is created between S and D', instead of between S and D. In this case, all subsequent data traffic generated by S will go to the attacker D', instead of D. We notice that the attack may fail when D is closer to S than D'. If the attacker D' and D are close enough to become neighbors, it is easy for D to know that it is attacked.

III. Methodology

In this section, we first describe a neighborhood-based method to detect the black hole attack and then present a routing recovery protocol to establish the path to the true destination such that the impact of the black hole attack can be mitigated. Although we use AODV as the example routing protocol to illustrate our method, our method could be slightly modified to be made applicable to the class of on-demand routing protocols, such as DSR.

A. Detection

In order to collect neighbor set information, we introduce two types of control packets in the detection phase: Request Neighbor Set (RQNS) and Reply Neighbor Set (RPNS).

The packet format of RQNS is as follows:

```
{Src.Addr, Dest.Addr, Request Neighbor Seq#, Next.Hop}
```

Src.Addr is the IP address of the source node S, and Dest.Addr is the IP address of the destination D. Each node is responsible for maintaining one counter: the sequence number of the RQNS. Each time a node sends a RQNS, Request Neighbor Seq# increases by one. The sequence number in each node uniquely identifies the RQNS, which unicast to the destination using the underlying AODV routing protocol.

D or D', after receiving RQNS, replies a message RPNS. The message format of RPNS is as follows:

```
{Src.Addr, Dest.Addr, Request Neighbor Seq#, Neighbor Set}
```

The first three items, i.e., Src.Addr, Dest.Addr, Request Neighbor Seq#, identify to which RQNS this RPNS corresponds. Neighbor Set contains the current neighbor set of D or D'. This RPNS unicasts back to S.

Based on this neighbor set information, we design a method to deal with the black hole attack, which consists of two parts: detection and response.

A. Detection

In order to collect neighbor set information, we introduce two types of control packets in the detection phase: Request Neighbor Set (RQNS) and Reply Neighbor Set (RPNS).

The packet format of RQNS is as follows:

```
{Src.Addr, Dest.Addr, Request Neighbor Seq#, Next.Hop}
```

Src.Addr is the IP address of the source node S, and Dest.Addr is the IP address of the destination D. Each node is responsible for maintaining one counter: the sequence number of the RQNS. Each time a node sends a RQNS, Request Neighbor Seq# increases by one. The sequence number in each node uniquely identifies the RQNS, which unicast to the destination using the underlying AODV routing protocol.

D or D', after receiving RQNS, replies a message RPNS. The message format of RPNS is as follows:

```
{Src.Addr, Dest.Addr, Request Neighbor Seq#, Neighbor Set}
```

The first three items, i.e., Src.Addr, Dest.Addr, Request Neighbor Seq#, identify to which RQNS this RPNS corresponds. Neighbor Set contains the current neighbor set of D or D'. This RPNS unicasts back to S.
Now we discuss our detection procedure. There are two major steps:

Step 1: Collect neighbor set information. Using AODV protocol, the source node S floods RREQ packets across the network to find a route to the destination node D. For each received RREP, S will unicasts a RQNS packet, and the RQNS packet will go to either D or D', depending on the path contained in RREP. After D or D' receives RQNS, it will generate a RPNS packet, which contains its current neighbor set, and unicasts it back to S. Figure 2 shows the sequence of the control packets among S, D, D' and intermediate nodes (IN).

Fig. 2. Time sequence of the control packets.

Step 2: Determine whether there exists a black hole attack. The source node S, after receiving more than one RPNS packet in a certain period, will start comparing the received neighbor sets. The difference among the neighbor sets is defined as the union of the received neighbor sets minus the intersection of the neighbor sets. If the difference is larger than the predefined threshold value, S will know that the current network has black hole attacks and takes some actions to respond to it.

One concern is that what if D' first requests the neighbor set of D, and replies it to S? We think that it is difficult for D' to do so. Because D' claims D's address, D' has to use D's address to request D's neighbor set, otherwise, D' s neighbors can find that D' is a masquerader. But D will raise an alert to this request, because it uses the same address of D.

B. Response

We assume there exists a public key infrastructure, which S can use to authenticate D or D'. After S detects the black hole attack, it will use the cryptography-based method to authenticate D and D'. In this way, S can identify D, the true destination. Note that, we only use this cryptography-based authentication method to identify the true destination after our neighbor set based method has detected that there exists a black hole attack in the network. This reduces the number of encryption/decryption operations.

Once D is identified, S will send a Modify.Route.Entry (MRE) control packet to D to form a correct path by modifying the routing entries of the intermediate nodes from S to D. We call this routing recovery protocol. The packet format of MRE is as follows:

\[ \text{Dest.Addr, Correct.Path} \]

\text{Dest.Addr} is the IP address of D, \text{Correct.Path} is the hop by hop path from S to D. S can get the information \text{Correct.Path} from the received RPNS. After each node receives the MRE, it will modify its corresponding routing entry (identified by the IP address of D) to make its next hop on the path to D, instead of D'. After D receives MRE, a correct path has formed between S and D, which will make the traffic of S go to the correct destination.

IV. Simulation

In this section, we describe our simulation environment and report the simulation results.

A. Simulation Setup

We use ns-2 [8] to build our simulation environment. We add the functions of our protocol to AODV to detect the black hole attacks. The Hello protocol is enabled because it can be used for each node to keep its neighbor sets. To run the simulations, we use PCs (600 MHz Celeron with 128M of RAM) running Red Hat Linux 7.1.

We assume that there is a one-to-one relationship between D and D', i.e., for each D, there is a corresponding D' who wants to intercept its data traffic. That is, if D' receives the RREQ, it generates RREP using the IP address of D. After D' receives the data traffic intended to D, it simply drops it for the purpose of simulation.

A.1 Performance Metrics

We use the following metrics to evaluate our protocol:

- Average Detection Time (ADT): This is the average time to detect that the network has black hole attacks. It is measured by the attack detection time minus the traffic start time.
- Control Packet Overhead: The overhead of routing control packets to detect the attack is introduced by RQNS and RPNS packets. We use the number of RQNS and RPNS packets to measure the overhead of our protocol.

Throughput: This is the percentage of the data packets sent that are actually received by the destination. We measure the throughput of the true destinations in two cases: when the routing recovery protocol is disabled and when the routing recovery protocol is enabled. They are compared to the throughput when there is no attack in the network.

Detection Probability: This is the probability that the attacker can be detected.

False Positive Probability: False positive occurs when S detects an attack while actually no attacker wants to intercept its traffic. In a certain period, false positive probability is the probability with which S detects attacks while there is no attacker.

False Negative Probability: False negative occurs when S does not detect the attack while actually there are attackers. In a certain period, false negative probability is the probability with which S does not detect attacks.

A.2 Movement and Communication Model

We use the distributed coordination function (DCF) of IEEE 802.11 for wireless LANs to simulate the MAC layer. We
set the traffic source to CBR (continuous bit rate), and the source-destination pairs spread randomly over the network. In our simulation, we use the following two field configurations: 1500*300 m field with 30 nodes and 1500*300 m field with 50 nodes. We want to measure the impact of the node density on our protocol. For each field configurations, we use ten connections. The CBR traffic is generated with a rate of 4 packets per second.

We use the random waypoint model[12] as the mobility model and generate two sets of moving scenarios. The first is that we change node pause time from the highest mobility (i.e. zero pause time) to 300 seconds; the second is that the maximum moving speed is changed from 2 m/s to 25 m/s. The pause time is varied to affect the relative speeds of the mobiles. We pre-generate 60 movement scenarios and run the simulation 800 seconds, in order to get the average effectiveness.

B. Simulation Result and Discussion

B.1 Average Detection Time

As we can see from the Figure 3, the ADT decreases when the maximum moving speed increases, and the ADT increases when the node pause time increases. For the same move pattern, there isn’t much difference of ADT between the two field configurations.

When the maximum moving speed is the changing parameter of the move patterns, we can see that the faster the maximum moving speed, the smaller the time used to detect the black hole attack of the network. Only if the traffic source receives two RPNS coming from both D and D' is it possible for S to detect the attack. The faster moving speed, the easier the link breaks, the more often RREQs broadcast, thus it takes less time to detect the attack in the current networks.

ADT increases when the node pause time increases. The pause time affects the relative speeds of mobiles, and they are not independent. If nodes tend to stop longer time after they reach their destinations, the links are harder to break. Therefore, it takes longer time for S to detect the attack.

When the move pattern is the same, the difference of ADT between two field configurations is small. This shows that the node density does not have much impact on the efficiency of our protocol.

B.2 Throughput

The throughput under various kinds of situations is shown in Figure 4 and 5. We do not want to perform a detailed routing protocol performance comparison such as [12]. We want to show the performance improvement of our protocol. In each figure, we plot five curves and their meanings are as follows:

- percentage of traffic to the destination when there are attacks and no routing recovery protocol (Dest recvd, Resp=OFF Atk=ON).
- percentage of traffic to the destination when there are attacks and routing recovery protocol (Dest recvd, Resp=ON Atk=ON).
- percentage of traffic to the destination when there are not attacks (Dest recvd, Atk=OFF).
- percentage of traffic to the attacker when there are attacks and no routing recovery protocol (Attacker recvd, Resp=OFF Atk=ON).
- percentage of traffic to the attacker when there are attacks and routing recovery protocol (Attacker recvd, Resp=ON, Atk=ON).

As we can see, when the attack exists in the network and the routing recovery protocol is disabled, the traffic which delivers to the attacker and which delivers to the destination is almost the same. We next plot improved throughput, i.e., the throughput to the destination when the recovery protocol is enabled. Simulation result shows that it will improve the throughput by at least 15%, depending on the movement scenarios. The top curve in these graphs is the throughput without attacks and with the enabled Hello protocols. There is still a gap between it and the improved throughput. The source node cannot detect the attack every time it initiates the route discovery, this will lead to the incorrect delivery of some traffic and is the main reason for the gap.
TABLE I

| COMPARISON OF THE NEWLY INTRODUCED CONTROL PACKETS (RQNS + RPNS) VS. TOTAL CONTROL PACKETS (RREQ + RREP + RQNS + RPNS) |
|------------|----------------|----------------|----------------|----------------|----------------|----------------|
| P  | RQNS+RPNS | RREQ+RREP+RQNS+RPNS |
| 0  | 42          | 695            |
| 30 | 56          | 666            |
| 60 | 52          | 758            |
| 100| 56          | 791            |
| 150| 56          | 840            |
| 200| 52          | 945            |
| 250| 53          | 849            |
| 300| 52          | 973            |

These data are for the 30-node network with the node pause time as the changing parameter of the node move patterns. Here $P$ is the node pause time in seconds. We can see that our detection protocol does not introduce much control overhead compared to the total number of control packets.

B.5 False Positive Probability

In order to measure the false positive probability of our method, we disable all of the attackers in the simulation environment, but still enable the detection part of our protocol. The simulation result is shown in Table III, where $P$ is the node pause time in seconds, $N$ is the number of nodes in the field and $S$ is the maximum moving speed in m/s. We can see that the false positive probability is very low, usually less than 1.7%.

<table>
<thead>
<tr>
<th>FALSE POSITIVE PROBABILITY VS. NODE PAUSE TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
</tr>
<tr>
<td>N=30</td>
</tr>
<tr>
<td>N=50</td>
</tr>
</tbody>
</table>

B.6 False Negative Probability

We measure the probability with which $S$ does not detect the attacks. The result is shown in Figure 8. Simulation shows that the larger the pause time, the larger the false negative probability. This is because when the pause time becomes larger, it is harder to detect the attacks.

![Figure 6](image1.png)

Fig. 6. Detection probability vs. node pause time.

![Figure 7](image2.png)

Fig. 7. Average number of control overhead used to detect the attack vs. node pause time.

We can also see that when the node pause time is the changing parameter, the throughput increases when the node pause time increases; when the maximum moving speed is the changing parameter, the throughput decreases when the node maximum moving speed increases. There is not much difference between the two different field configurations, i.e., the node density does not have much impact on the performance of our protocol.

B.3 Detection Probability

As shown in Figure 6, the detection probability increases as the maximum moving speed and the node pause time increases. When the nodes move faster, the chance with which the path breaks increases. Thus, it is more often for the source node to initiate the route discovery procedure and the black hole attack is more likely to be detected with our methods. We also observe that with the increasing node pause time, the detection probability decreases. The reason is similar.

B.4 Overhead

The result of the routing control overhead is shown in Figure 7. As we can see, for both field configurations, the control overhead does not change much throughout the move patterns, and doesn’t show an obvious trend when maximum moving speed or node pause time changes. The number of $RQNS$s is always larger than the number of $RPNS$s. This is because the $RPNS$ packet is generated by the destination when it receives a $RQNS$ packet, and the $RQNS$ packet can possibly get lost during its transmission.

Table II shows the comparison of the newly introduced control packets ($RQNS + RPNS$) vs. total control packets ($RREQ + RREP + RQNS + RPNS$). These data are for the 30-node network with the node pause time as the changing parameter of the node move patterns. Here $P$ is the node pause time in seconds.

![Figure 8](image3.png)

Fig. 8. False negative probability.
From simulation results, we can see that our proposed methods can effectively detect the black hole attack in terms of acceptable detection probability without introducing too much control overhead into the network.

V. Related Work

Up to now, almost all of the mobile ad hoc routing protocols lack built-in security. Retrofitting security mechanisms into these routing protocols are often expensive.

In [7], a routing misbehavior in mobile ad hoc networks is described. The authors proposed an effective method to improve the throughput in the presence of the misbehaving nodes. By setting each node to the promiscuous mode, the proposed method installs two facilities: the watchdog and the path tracer. This method can be applied to address a special form of black hole attack (i.e., the malicious node intentionally drops the packet). However, it is not effective to address the problem we discussed in this paper.

VI. Conclusions and Future Work

In this paper, we proposed a neighbor set based approach to detect black hole attack and a routing recovery protocol to mitigate the effect of black hole attacks. We demonstrated through simulation that our methods could effectively and efficiently detect black hole attack without introducing much routing control overhead to the network. Simulation data shows that the packet throughput can be improved by at least 15% and the false positive probability is usually less than 1.7%. In the future, we would like to further explore whether there exists a non-cryptography based method to identify the true destination and the optimal detection and response mechanism to improve the packet throughput.

REFERENCES