An Acknowledgment-Based Scheme to Defend Against Cooperative Black Hole Attacks in Optimized Link State Routing Protocol

Soufine Djahel$^{\mp}$, Farid Naït-Abdesselam$^{\mp}$ and Ashfaq Khokhar$^{\pm}$

$^{\mp}$LIFL/IRCICA – CNRS UMR 8022
University of Sciences & Technologies of Lille, France
{farid.nait-abdesselam, soufiene.djahel}@lifl.fr

$^{\pm}$Electrical and Computer Engineering Department
University of Illinois at Chicago
ashfaq@ece.uic.edu

Abstract— **In this paper, we address the problem of cooperative black hole attack, one of the major security issues in mobile ad hoc networks. The aim of this attack is to force nodes in the network to choose hostile nodes as relays to disseminate the partial topological information, thereby exploiting the functionality of the routing protocol to retain control packets. In optimized link state routing (OLSR) protocol, if a cooperative black hole attack is launched during the propagation of topology control (TC) packets, the topology information will not be disseminated to the whole network which may lead to routing disruption. In this paper, we investigate the effects of the cooperative black hole attack against OLSR, in which two colluding MPR nodes cooperate in order to disrupt the topology discovery. Then we propose an Acknowledgment based technique that overcomes the shortcomings of the OLSR protocol, and makes it less vulnerable to such attacks by identifying and then isolating malicious nodes in the network. The simulation results of the proposed scheme show high detection rate under various scenarios.**

Keywords – **Ad Hoc Networks, Routing Protocols, OLSR, Security, Cooperative Black Hole.**

I. Introduction

In many applications, mobile ad hoc networks may be deployed in a hostile environment. Due to numerous constraints such as, lack of infrastructure, dynamic topology and lack of pre-established trust relationships between nodes, most of the envisioned routing protocols are vulnerable to a number of disruptive attacks. In this paper, we focus on the cooperative black hole attack which is known to be particularly challenging to defend against [5], and has been shown to be potentially damaging to a wide range of ad hoc routing protocols.

In black hole attacks, hostile nodes in general advertise availability of a fresh route without checking their routing tables. In this process, attackers always happen to be the first to reply to a route request in reactive routing context, and thus intercept the data packets being relayed and retain them. Furthermore, a special case of black hole attack, called gray hole attack, is mentioned in [4] in which only part of packets are retained while the other part is relayed. In the literature [5], [9], there are similar definitions of black and gray hole attack. However, a gray hole attack is considered as a special case of the black hole attack that has a similar impact but more difficult to detect. When this attack targets a proactive routing protocol, such as OLSR, where attackers may retain topology control packets, nodes in the entire network would be unable to get the right picture of the network topology. This will result in disrupting severely the communications as attackers will always be chosen by their neighbors as potential next hops to the destinations.

In this paper, we introduce an efficient method to detect cooperative black hole attacks. It uses an acknowledgment approach to ascertain the effective dissemination of topology control packets within the network. First, each MPR (MultiPoint Relay) node have to learn about its neighborhood up to three hops away, and then it piggybacks in its next TC message a request to an acknowledgment only from a selected subset of this set. Upon reception of the TC message, each node from this subset responds by an authenticated acknowledgment to confirm the reception of the identified TC message. By monitoring the number of missing acknowledgments, each MPR node can verify whether a cooperative black hole attack is carried out in its vicinity or not. To mitigate the impact of this attack, the detected malicious nodes will be excluded from the forwarding process by simply ignoring their Hello packets.

The rest of the paper is organized as follows. The next section summarizes the literature. Section III provides a short overview on OLSR, followed by the description of a cooperative black hole attack in section IV. In section V, we present the proposed acknowledgment based scheme to detect the misbehaving nodes. Section VI describes our simulation model and analyzes the obtained results. Finally, section VII discusses future research directions and concludes the paper.

II. Related Work

Recently, a number of protocols and mechanisms have been proposed to secure wireless ad hoc routing. In [3], a new protocol called CONFIDANT is proposed, which aims at detecting malicious nodes by means of combined monitoring, reporting and establishment of routes that avoid misbehaving nodes. In this scheme, events have to be observable and classifiable for detection, and reputation can only be meaningful if the identity of each node is persistent, otherwise it presents vulnerabilities to spoofing attacks.

Another approach is proposed in [5] to defend against black hole attack in AODV, where they use *further request* and *further reply* packets to check the validity of a node that sends the route reply before sending the data packets. This solution assumes that malicious nodes do not exist in groups, although this is quite possible in real situations.

[6] proposes a neighborhood based method to defend against black hole attack. The solution can be briefly described as follows: once the normal path discovery procedure in the routing protocol is done, 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,

they designed a routing recovery protocol to establish the path to the correct destination.

The most effective and recent method to defend against black hole attacks is the watchdog/pathrater mechanisms[2]. This method detects misbehaving nodes acting alone by maintaining a buffer that contains recently sent packets. When a node forwards a packet, the node's watchdog ensures that the next node in the path also forwards the packet. The watchdog does this by listening to the next node promiscuously. If the next node does not forward the packet then it is termed as misbehaving. In other words, in this scheme, every packet that is overheard by the watchdog is compared with the packet in the buffer to see if there is a match. A match confirms that the packet has been successfully delivered and it is removed from the buffer. If a packet has remained in the buffer beyond the timeout period then a failure counter for the node responsible for forwarding the packet is incremented. If this counter exceeds a pre-determined threshold then the node is termed as malicious and the network is informed accordingly.

In [10], the authors present a new collusion attack model against OLSR, where the first colluding node declares in its Hello message all the 2 hop neighbors of the attacked node in order to force this latter to choose it as its unique MPR. The second attacker, acting as MPR of the first attacker, will then drop all the packets origintaed from the attacked node and passing through it. To defend against this attack, they propose to modify the standard Hello message to contain the 2 hop neighbors list. Based on this information a node can detect whether one of its neighbors has sent a false Hello message or not by looking at contradictions in the sets. However, these contradictions can no more stand if the attacker sends false neighbors set which are different from the legitimate node's 2 hop neighbors.

III. OLSR DESCRIPTION

The Optimized Link State Routing protocol (OLSR) [7] is a proactive routing protocol designed to work in dense networks. The main optimization of the protocol is achieved through MPRs, a set of neighbor nodes that are unique nodes in the network responsible for generating and distributing partial link state information during the flooding process, thus reducing the message overhead.

In OLSR, each node selects its MPR set from its 1 hop neighbors such that it can reach easily all its 2 hop neighbors. MPR selection process depends on the 'Willingness' value obtained from Hello message. This value indicates the willingness of a node, based on its own resources, if it is able to forward packets of other nodes. The higher the willingness is, the higher the priority the node will have to be selected as MPR. The value $Will\_never$ is chosen by nodes that are not willing to participate in the routing process. However, the value $Will\_always$ is reserved for nodes which are candidate to be selected as MPRs.

Control traffic in OLSR is exchanged through two different types of messages, namely 'Hello' and 'TC' messages. Hello messages are exchanged periodically (every 2s) between nodes to detect links between each others, to detect the identity of neighbors and to advertise MPR selections. A TC message is sent to the whole network's nodes periodically by each MPR node to declare its MPR selectors set. Information contained in TC message is used in the construction of the routing tables in each network's node.

IV. COOPERATIVE BLACK HOLE ATTACK MODEL

In this section, we describe how two adjacent malicious nodes can launch a black hole attack in wireless ad hoc network. To ease understanding the illustration, we summarize bellow the notations and assumptions used throughout this paper. First, we assume that links are frequently symmetric. We assume also that each node holds a set of secrets for all other nodes. For example, the secret K_{ij} , stored within a node i, is the secret associated to node j. The secret is assumed to be symmetric between any pair of nodes, i.e, $K_{ij} = K_{ji}$, and are undisclosed to any other node in the network. The following notations are also used to illustrate the cooperative black hole attack in OLSR.

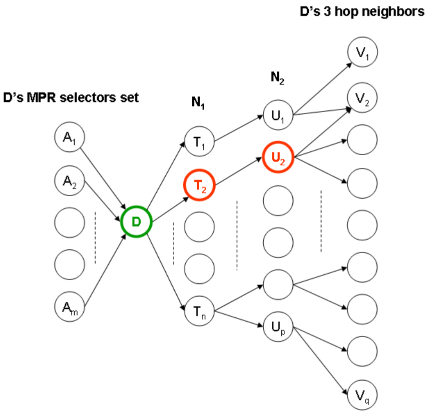

Figure 1: A cooperative attack model

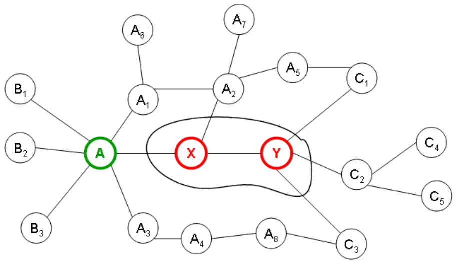

Figure 2: Cooperative black hole attack description

- N_1: the MPR set of node D.
- N_2: the MPR set of N_1's nodes which are not in D's one hop neighborhood, i.e.,

$$\forall n \ (n \in N_2 \Rightarrow \exists m \ (m \in N_1 \ \wedge T))$$

$$T \equiv n \in MPR\_set\,(m) \wedge n \notin neigh\,(D)$$

- Sym_i: the symmetric 3 hops away neighbors which have to send out an $3hop\_ACK$ packet.
- S_i: the subset of symmetric neighbors 3 hops away reached through the node $n \in N_2$.

In order to launch the black hole attack in OLSR, a malicious node can force its election as MPR by maintaining constantly its Willingness field to $Will\_always$ in its HELLO message. According to the protocol, its neighbors will always select it as MPR. Using this mechanism, a malicious node can easily earn, as an MPR, a privileged position within the network. It can then exploit its rank to carry out deny of service attacks and alike. In a more sophisticated way, two colluding MPR nodes m_1 and m_2 can launch a more severe attack when the node m_2 drops all TC messages forwarded by node m_1. The attacked node, in the set of MPR selectors of m_1, can not detect this misbehavior because node m_2 is out of its radio range.

Fig. 1 shows an illustrative description of this cooperative black hole attack. Let nodes $\{A_1,..., A_m\}$ be a set of nodes to be attacked and T_2, U_2 the attacker nodes, $\{T_1,..., T_n\}$ the set of D's MPR nodes, $\{U_1,..., U_p\}$ is the subset of D's 2 hop neighbors which constitutes the N_1's MPR nodes and $\{V_1,..., V_q\}$ the set of D's 3 hop neighbors. The attack is launched as follows: node T_2 sends its HELLO message with the value of $Willingness$ field as $Will\_always$, i.e, all its 1 hop neighbors will choose it as an MPR. Then it chooses the node U_2 as the only MPR node to relay its TC message. By doing this,

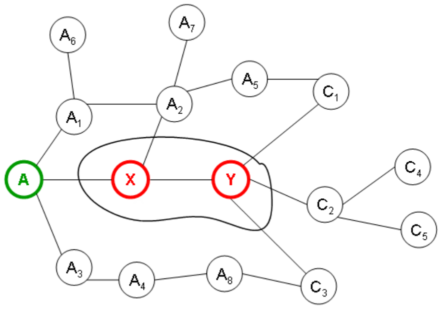

Figure 3: Topology perceived by nodes C_2, C_4 and C_5 after attack

node U_2 can perform the following misuses without being detected by node D or any other node in its neighborhood:

- Drop all TC messages generated by node D, i.e, which contains addresses of nodes $\{A_1,...., A_m\}$. This will avoid the links status of D's MPR selectors to reach the nodes that are only connected to U_2. This means that some nodes in the network can not communicate with the D's MPR selectors set.
- Drop TC messages generated by node T_2 which contains the address of D to prevent link information of D from being disseminated to the nodes reached through it. It can also drop all TC messages passed through it.
- Modify the content of TC messages generated by node D to cause inconsistent topology information dissemination.

The consequence of this attack is devastating in the presence of multiple colluding attackers around the victim node. The sender/forwarder of TC message overhears its MPR nodes transmission to ensure whether or not its sent/forwarded TC is relayed, but while the TC dropping happens out of the sender/forwarder transmission range this technique failed to detect it.

Fig. 2 shows an example of this attack. Nodes B_1, B_2 and B_3 constitute the MPR selectors set of node A. The first attacker X advertises itself as having the sufficient resources to forward packets of other nodes by setting its willingness field to the highest allowed value (i.e, the value 7). According to the protocol, X will be chosen as the A's MPR. Afterward, X chooses the second attacker Y as its unique MPR node. Thus, all TC packets generated by node A and relayed by X will be destroyed by Y.

The consequence of this attack is illustrated in Fig. 3, where nodes C_2, C_4 and C_5 can not build a route toward D's MPR selectors because the D's TC messages are never received (i.e, the topology information held by these nodes is incomplete).

V. The proposed solution

In order to deal with the cooperative black hole attack, we present an acknowledgment based scheme to mitigate the loss of topology information due to the dropping of TC messages by attackers. In addition to the original control message of OLSR, i.e, Hello and TC messages, our scheme introduces another two kinds of control packets, which are called $3hop\_ACK$ and $HELLO\_rep$. The $3hop\_ACK$ message is used by a node to acknowledge its reception of a TC message from the neighbors 3 hops away, and the $HELLO\_rep$ message is used by a node to advertise its 2 hop neighbors to a requesting MPR node. For the request, we use one of the unused bits in the HELLO message to indicate whether the sender's MPR nodes should generate $HELLO\_rep$ packet or not. Table I shows the format of $HELLO\_rep$ message.

Our scheme requires that each MPR node should know its 3 hop neighbors set in order to be able to verify whether a malicious node, sitting out of its transmission range, misuses the transmitted TC messages. Our scheme also requires that each MPR node has to forward all TC messages from its MPR selectors even if a message is received more than once. As such, the scheme can distinguish whether a TC message is dropped intentionally by a malicious node or by a legitimate node just because of the duplication of two TC messages.

MPR_node	$2\_hops\ neighbors$
Y	C_1 C_2 C_3
A_2	A_1 A_5 A_7

Table I: Example of $HELLO\_rep$ message sent by node X

To detect the misbehaving nodes, the sender or forwarder of the TC message maintains a list of TC packet identifiers (IDs) that have not received any $3hop\_ACK$ packet from symmetric neighbors 3 hops away. Also, a number of parameters need to be maintained, as follows:

- ACK_{miss}: Counter for the number of $3hop\_ACK$ missed on the link $M_1 \longleftrightarrow M_2$ such that $M_1 \in N_1$ and $M_2 \in N_2$.
- TC_{drop}: Counter for the number of TC dropped by the direct MPR node.
- SNT: List of nodes which should send out a $3hop\_ACK$ for each pair of nodes $<M_1, M_2>$.
- δ_1 and δ_2: Threshold for TC packets dropped and $3hop\_ACK$ packets missed respectively ($\delta_1 < \delta_2$).
- $BlackList$: List of misbehaving nodes.

When a node, say, A sends or forwards a TC message to its neighbors, it monitors its MPR's transmission to verify whether MPR nodes relay the packet or not. If A doesn't overhear that of MPR node B, it increases its TC_{drop}. If TC_{drop} exceeds δ_1, A adds B's identity to the $BlackList$ and recalculates the MPR set without node B. Another more complicated case is the case where node B forwards the TC message sent by A, while it doesn't punish its MPR node C which drops the TC message. To deal with this case, node A increases the ACK_{miss} counter for the link $B - C$ whenever the timer, which is set up to limit the reception delay of $3hop\_ACK$ packets for this link, exceeds the predefined threshold. If node A receives a Hello message from B in which C is not in B's MPR set, while ACK_{miss} does not exceed δ_2, A then adds C to its $Suspicious_{list}$; otherwise, node A concludes that both nodes B and C are malicious and adds them to its $BlackList$.

$Suspicious_{list}$ is used by a MPR node A to record the IDs of nodes that drop the Acknowledgment packets but move out of the forwarding path of TC messages before their Ack_{miss} counter reaches the threshold. As the network's topology changes rapidly, suspicious nodes can gain the same position in the forwarding path of TC messages sent/forwarded by the node A. A then accumulates the new observations with the previous ones for a more accurate punishment.

On receiving Hello or TC message, each node deals with it as algorithms 1 and 2.

Our scheme is summarized as follows:

1) First step: Get all neighbors 3 hops away
   - For each node i of the N_1 set, node D implicitly asking it to send its MPR set (i.e, MPR nodes of i) and the neighbors of these MPRs (i.e, 2 hops neighbors of i reached through these MPRs).
   - Upon reception of $HELLO$ message in which $HELLO\_id$ is set to 1, each node of the N_1 set responds with a $HELLO\_Rep$ packet containing the required information.
   - Each MPR node sets the $HELLO\_id$ field to 1 whenever it detects a change in its MPR set or its N_2 set.
2) Second step: Sending or forwarding TC message with request of acknowledgment

Algorithm 1 HELLO reception

```
if $orig\_adr \notin BlackList$ then
    if $Id = 1$ then
        if $orig\_adr \in MPR\_sel\_set$ then
            $prepare$ HELLO_rep pkt
            $send$ HELLO_rep pkt
        end if
        $process$ HELLO msg
    else
        $process$ HELLO msg
    end if
end if
```

Algorithm 2 TC reception

```
if ($orig\_adr$, PSN)$\notin$ duplicate_set then
    if $my\_Id \in Req\_ack\_list$ then
        $prepare$ 3hop_ack_pkt
        $send$ 3hop_ack_pkt
    end if
    $process$ TC
else
    if sender_addr $\in$ MPR_selc_set then
        $add$ its own $Request\_ack\_list$ to $TC\_pkt$
        $forward\ TC\_pkt$
    end if
end if
```

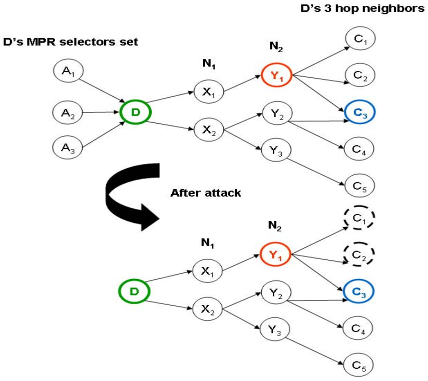

Figure 4: Topology perceived by nodes C_1 and C_2 when the condition c is not satisfied

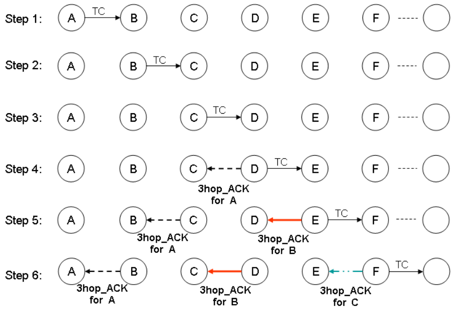

Figure 5: The $3hop\_ACK$ scheme functioning

- For each node of the N_2 set, the node D chooses a neighbor node such that:

$$\left.\begin{array}{l} \forall i \;\; sym_i \;\notin \bigcup_{\substack{j=1,\dots,p\\k=1,\dots,p}}^{j\neq k} (\; S_j \cap S_k \;) \\ p \;=\; \|N_2\| \end{array}\right\} \quad (\,c\,)$$

- When the node D generates its TC message or forwards a TC message received from its MPR selectors, it inserts a request for an acknowledgment to all nodes of the set X, before sending it out.

$$X = \{Sym_1, ..., Sym_p\}$$

- Since PKI-based authentication is not suitable to MANETs due to the high computational complexity, we adopt an alternative solution proposed in [8] to validate the identity of $3hop\_ACK$ sender, which has moderate computation complexity and low bandwidth consumption.

A. Discussion

In Fig. 4, node D may choose node C_3, where $C_3 \in (S_{Y_1} \cap S_{Y_2})$, to send back a $3hop\_ACK$. In this case, node C_3 can be reached through both Y_1 and Y_2 of the set N_2. We suppose that Y_2 is a legitimate node while Y_1 is a misbehaving node. Since D maintains a list of nodes that should send $3hop\_ACK$, when node C_3 receives a TC message forwarded by Y_2, it sends a $3hop\_ACK$ to D. When D gets this acknowledgment, it will delete C_3 from the SNT list. Thus node D believes that both nodes Y_1 and Y_2 have relayed its TC message, but actually node Y_1 did not forward it. Even node Y_1 forwards the TC message, node C_3 sends back only one $3hop\_ACK$ because it processes only the first TC message it received and ignores the later one that has the same *originator address* and *MSN*(Message Sequence Number).

B. The $3hop\_ACK$ scheme operation

Since our scheme works at the network layer, it can be implemented as an adds on to OLSR protocol. The scheme detects individual malicious nodes by using a new type of Acknowledgment packets, named $3hop\_ACK$. The $3hop\_ACK$ packet is assigned a route of 3 hops in the opposite direction of TC packet route as shown in Fig. 5.

In Fig. 5, A is assumed to be an MPR node, $B \in N_1$, $C \in N_2$ and D the A's 3 hops neighbor which should send back the $3hop\_ACK$. When node A detects changes in its N_1 or N_2 set, it sends a HELLO message with the $HELLO\_id$ field set to 1. Upon reception of this message, each node from the set N_1 sends a $HELLO\_Rep$ packet to A. When TC_timer expires, node A generates its TC message along with a request for acknowledgment to the node D. This message will be forwarded by B and C respectively. Each receiver of this TC message checks whether its identity is included in the request list for acknowledgment; if so, it returns an authenticated $3hop\_ACK$ packet (as shown in Fig. 5). When node A receives such packet, it decrypts it using the shared secret key with the anticipated sender, verifying whether the packet is sent by a legitimate node or not. Finally, node A concludes that the two consecutive MPR nodes in this forwarding path behave well. This process is repeated for every quadruplet of nodes as $< B, C, D, E >$, $< C, D, E, F >$ and $< D, E, F, ... >$, where the first 3 nodes are MPR nodes and the last one may not be an MPR node.

C. Security analysis of $3hop\_ACK$ scheme

Our scheme is robust to a variety of attack scenarios conducted by either independent or collusive malicious nodes. In particular, we consider three typical scenarios as follows:

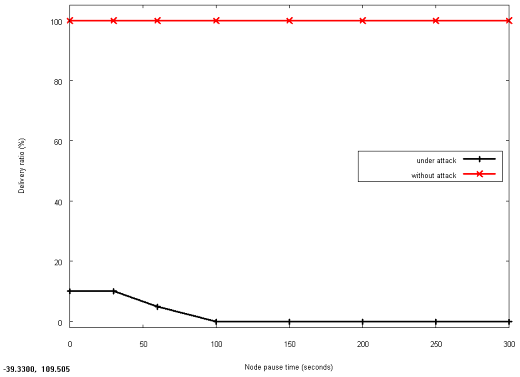

Figure 6: Delivery ratio vs. node pause time

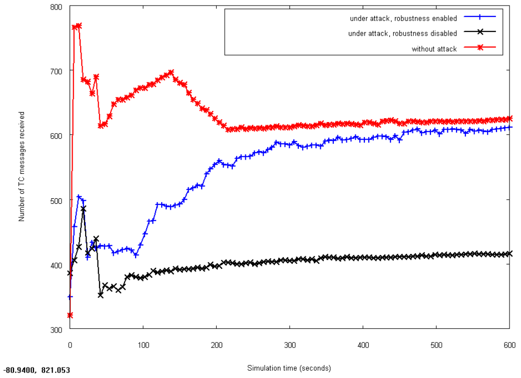

Figure 7: Comparaison of the average number of TC messages received

- Scenario 1: a node in N_1 set responds with a faulty list of 2 hop neighbors in which it adds non neighbor nodes, or deletes some nodes from this list. According to our scheme, when node D receives the $Hello\_rep$ from node X it chooses one of the nodes listed in the 2 hop neighbors list, requiring it to send a $3hop\_ACK$. If the selected node is not an actual 2 hop neighbors of X, D would not receive a $3hop\_ACK$. As a result, node X will be punished.
- Scenario 2: a node in N_2 set drops the TC message and fabricates a $3hop\_ACK$ packet on behalf of the requested node. In order to fabricate this packet, the node must spoof the requested node identity and generates a valid response. But it could not compute the valid $3hop\_ACK$ due to the lack of knowledge on the secret key pre-shared between the requester and requested nodes.
- Scenario 3: a node $X_2 \in N_2$ relays the TC message correctly but drops the $3hop\_ACK$ packet. This behavior would be detected as well, as the node $X_1 \in N_1$ monitors its behavior and deletes it from the 1 hop neighbor list when the ACK_{miss} counter reaches the threshold δ_2.

VI. Simulation model and results

This section reports the performance evaluation on our scheme by using extensive simulations conducted with the network simulator OPNET 11.5 [11]. We embedded our scheme in the implemented OLSR protocol for detecting the cooperative black hole attack. We generated random topologies with M nodes over a rectangular field, where M ranges from 30 to 100. The rectangular field size is varied from (1500x1000m) to (2500x2000m). The maximum transmission range of each node is 250m. We run the simulation for 600s. Random waypoint model [1] is used as the mobility model of each node. Nodes speed is varied from 2 m/s to 25 m/s. We change node pause time from the highest mobility (i.e, zero pause time) to 300 seconds. We use the default setting for Hello and TC messages generations as in the specification of OLSR [7]. The percentage of malicious nodes is varied from 25% (which launches 4 cooperative black hole attacks) to 40% (which launches 20 cooperative black hole attacks). To launch the attack, the first attacker chooses randomly a victim node from its MPR selector set that has to be an MPR of the other neighbors.

A. Cooperative Black hole attack simulation

To illustrate the consequence of this attack, we generated traffic between a source node and a destination node where they are more than 3 hops away. In our scenario, this chosen destination node has as its MPR the attacked node. We define also the delivery ratio as a value of the number of received data packets to that of packets being sent by the source node. The results are shown in Fig. 6, and we observe that in the presence of the attack, the delivery ratio is 10% when node pause time is 0 and 30 seconds and it drops to 5% when node pause time goes to 60 seconds. The reason is that, when the destination node moves faster, it has more chances to select nodes other than the victims as MPRs and thus receives more packets; while with the lower mobility, it forces the destination node to select victim nodes as MPRs, thereby leading to fewer routes to be established between the source node and the destination.

B. Performance evaluation

We conducted extensive simulations to evaluate the performance of our scheme and show the results in Fig. 7, where the meaning of three curves are explained as follows,

- Under attack, robustness disabled: average number of receptions of each TC message generated by MPR nodes when there are attacks and our scheme is not used.
- Under attack, robustness enabled: average number of receptions of each TC messages generated by MPR nodes when there are attacks and our scheme is used.
- Without attack: average number of receptions of each TC messages generated by MPR nodes when there are no attacks.

Clearly, Fig. 7 shows that the average number of TC messages received by nodes decreases sharply when the network is under attacks and no robustness. In this case, some nodes have a partial image of the network topology because of the no reception of all TC messages generated by MPR nodes, therefore they could not establish routes to some other nodes.

When the attack is launched and our scheme is used, the average number of TC messages received is small at the beginning because the attacker node drops TC messages generated by victim nodes. This number increases at each time an attack is detected, and finally it approaches to the number of TC messages received when there is no attack. The implicit reason is that, once a malicious node is detected, the victim nodes elect a new MPR set of legitimate nodes which forwards the TC messages correctly.

Fig. 8 shows the relationship between detection rate and node pause time. We see that generally the detection rate increases as the node pause time turns to larger. Especially, when node pause time is 0, i.e., nodes move continuously, the MPR sets of nodes change rapidly, thereby malicious nodes can drop packets and move away before its failure counter reaches the threshold. In another word, the frequency of changing of MPR set decreases with the increase of pause time, so when pause time gets larger, MPR sets is more stable over time and the performance of our scheme gets better.

In order to observe the number of false alarms triggered by our scheme, we generate a network with random topology and consists of 100 nodes, among which 40% are malicious nodes. As shown in Fig. 9, Y axis is the number of false alarms, X axis is the timeout

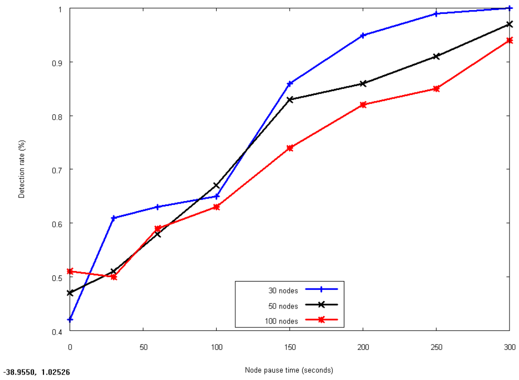

Figure 8: Detection rate vs. node pause time

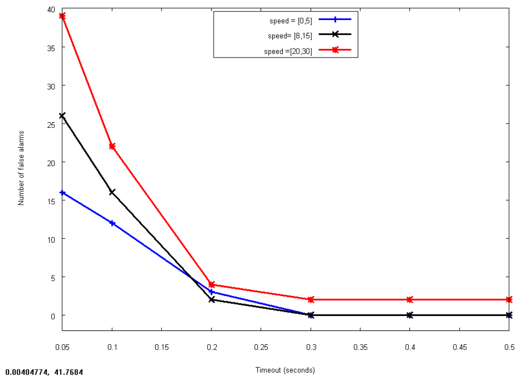

Figure 9: Number of false alarm vs. timeout value

value δ, and the maximum speed of nodes varies from 0 m/s to 30 m/s and is divided into 3 ranges. We observed that the number of false alarms is generally proportional to the nodes speed and reduces with the increasing timeout value. We also observe that $\delta = 0.3$ is the optimal threshold, after which false alarm never occur when mobility speed range is $[0, 5]$ and $[8, 15]$, while only 2 false alarms keep occurring for mobility speed range $[20, 30]$.

VII. Conclusion

Cooperative black hole attack is one of the most sophisticated attack that may result in dramatic disruption of the network performance. When launched against the OLSR protocol, a misbehaving node exploits the routing protocol's vulnerabilities to always participate in the forwarding process of TC messages, and when it colludes with another adjacent MPR node, they cooperate to prevent the dissemination of these TC messages generated by the original node in order to false the discovery of the network topology.

In this paper, we have analyzed the cooperative black hole attack in ad hoc networks and proposed an acknowledgment based scheme to detect malicious nodes and isolate them from the forwarding process. We conducted extensive simulations and the results showed high detection rate under various scenarios with negligible false positive alarms.

References

[1] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu and J. Jetcheva, "A performance Comparaison of Multi-Hop Wireless Ad HOc Network Routing Protocols", *in Proc. of the* $4th$ *annual ACM/IEEE international conference on Mobile computing and networking (ACM Mobicom '98)*, Dallas, Texas, USA, October 1998.

[2] S. Marti, T.J. Giuli, K. Lai and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks", *in Proc. of the* $6th$ *annual international conference on Mobile computing and networking (MOBICOM '00)*, Boston, Massachusetts, USA, August 2000.

[3] S. Buchegger and J.Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol", *in Proc. of the* $3rd$ *ACM international symposium on Mobile ad hoc networking & computing (MOBIHOC'02)*, Lausanne, Switzerland, June 2002.

[4] Y. Hu, A. Perrig and D. Johnson, "Ariadne: A secure On-Demand Routing Protocol for Ad Hoc Networks", *in Proc. of the* $8th$ *ACM international Conference on Mobile Computing and Networking* , Westin Peachtree Plaza, Atlanta, Georgia, USA, September 2002.

[5] H. Deng, W. Li and D.P. Agrawal, "Routing security in wireless ad hoc networks", *in IEEE Communication Magazine*, Vol. 40, No. 10, pp. 70-75, october 2002.

[6] B. Sun, Y. Guan, J. Chen and U.W. Pooch, "Detecting black- hole attack in mobile ad hoc networks", *in Proc. of the* $5th$ *European Personal Mobile Communications Conference*, Glasgow, UK, April 2003.

[7] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)". *IETF RFC 3626 (Experimental)*, October 2003.

[8] Y.R. Tsai and S.J. Wang, "Routing security and authentication mechanism for mobile ad hoc networks", *in Proc. of the* $60th$ *IEEE Vehicular Technology Conference*, Los Angeles CA, USA, September 2004.

[9] W. Yu, Y. Sun and K.R.Liu, "HADOF: Defense Against Routing Disruptions in Mobile Ad Hoc Networks", *in Proc. of the* $24th$ *IEEE INFOCOM*, Miami, USA, March 2005.

[10] B. Kannhavong, H. Nakamaya and A. Jamalipour, "A Collusion Attack Against OLSR-based Mobile Ad Hoc Networks", *in Proc. of Global Telecommunications Conference (GLOBECOM '06)*, San Francisco, California, USA, Nov/Dec 2006.

[11] OPNET Technologies. *OPNET Modeler.* http://www.opnet.com/.