www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 6 June 2017, Page No. 21780-21788
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i6.38

Mitigating Denial of Service Attacks in OLSR Protocol Using Fictitious Nodes

Nagaashwini Nayak V J , Nagaveni B Biradar
Assoc Prof, RYMEC
Email id:nagaashwininayak@gmail.com, veer_nagaveni@yahoo.com

Abstract—With the main focus of research in routing protocols for Mobile Ad-Hoc Networks (MANET) geared towards routing efficiency, the resulting protocols tend to be vulnerable to various attacks. Over the years, emphasis has also been placed on improving the security of these networks. Different solutions have been proposed for different types of attacks, however, these solutions often compromise routing efficiency or network overload. One major DOS attack against the Optimized Link State Routing protocol (OLSR) known as the node isolation attack occurs when topological knowledge of the network is exploited by an attacker who is able to isolate the victim from the rest of the network and subsequently deny communication services to the victim. In this paper, we suggest a novel solution to defend the OLSR protocol from node isolation attack by employing the same tactics used by the attack itself. Through extensive experimentation, we demonstrate that 1) the proposed protection prevents more than 95 percent of attacks, and 2) the overhead required drastically decreases as the network size increases until it is non-discernable. Last, we suggest that this type of solution can be extended to other similar DOS attacks on OLSR.

Index Terms—MANET, OLSR, node isolation attack, fictitious node

1 INTRODUCTION

A Mobile Ad Hoc Network (MANET) is a group of mobile devices capable of communicating wirelessly with each other without using a predefined infrastructure or centralized authority. Sending packets from one device to another is done via a chain of intermediate nodes.

A number of different routing algorithms exist for network packet transmission. For the most part these algorithms can be classified into two main categories: reactive routing and proactive routing protocols. In the case of proactive (table-driven) protocol, for example, DSDV and OLSR, every node constantly maintains a list of all possible destinations in the network and the optimal paths routing to it. Reactive protocols, such as DSR and AODV, find a route only on demand.

Irrespective of routing algorithm, one of MANET's essential requirements of and a factor in its success is its ability of having all nodes recognized by other participants, even in motion.

These algorithms differ from the standard routing used in classic networks due to frequent topology changes. A route between two nodes can be broken due to intermediate nodes that dynamically change their position. Mobile nodes can join or leave the network at will, further influencing network connectivity.

Of the routing protocols mentioned above a proactive algorithm, the Optimized Link State Routing (OLSR) protocol has become one of the algorithms widely used today. Although OLSR is quite efficient in bandwidth utilization and in path calculation, it is vulnerable to various attacks. As OLSR relies on the cooperation between network nodes, it is susceptible to a few colluding rogue nodes, and in some cases even a single malicious node can cause routing havoc. These attacks include link withholding attacks, link spoofing attacks, flooding attacks, wormhole attacks, replay attacks, black-hole attacks, colluding mis-relay attacks, and DOS attacks.

In this paper we review a specific DOS attack called node isolation attack and propose a new mitigation method. Our solution called Denial Contradictions with Fictitious Node Mechanism (DCFM) relies on the internal knowledge acquired by each node during routine routing, and augmentation of virtual (fictitious) nodes. Moreover, DCFM utilizes the same techniques used by the attack in order to prevent it. The overhead of the additional virtual nodes diminishes as network size increases, which is consistent with [4]'s general claim that OLSR functions best on large networks.

The reminder of this paper is organized as follows. In Section 2 the OLSR protocol is presented; then, the node isolation attack is described and other previous works related to OLSR MANET security are discussed. A method for protecting OLSR MANET from node isolation attack is described in depth in Section 3. Section 4 describes the simulation model and presents the results achieved along with a discussion of the results. Finally, conclusions and future works are presented in Section 5.

2 BACKGROUND AND RELATED WORK

2.1 OLSR Overview

The OLSR protocol is an optimization of the classical Link-State Routing protocol (LSR), aimed at reducing network overhead. While the original LSR uses a flooding propagation technique in which a node receiving any message must retransmit it to all its neighbors, OLSR selectively retransmits messages based on a specified set of rules. The crux of the optimization is based upon a subset of one-hop neighbors, called multi-point relays (MPR), which are designated as forwarding agents for control packets throughout the network.

MPRs are selected by a node as a subset of its one-hop neighbors, such that the MPR set allows coverage of all of its two-hop neighbors. By minimizing its MPR selections, a node is able to transmit messages to all two-hop neighbors with minimal duplication. Thus, both topology control messages and data packets are only forwarded by this minimal MPR set, allowing for fewer duplicate messages while maintaining network-wide coverage.

There are two types of messages used to discover network topology in OLSR: HELLO and TC (i.e., topology control). The HELLO message, which declares a node's knowledge of its surrounding, is broadcast to all. Any node that can hear the broadcast

and reciprocate back to the sender is classified as a one-hop neighbor. Consequently, each node acquires its local topology up to a two-hop range. In addition, OLSR requires that all nodes selected as MPRs periodically advertise a TC message listing all nodes that have selected the sender as its MPR. These control messages are only propagated through the MPR super-network, reducing overall network traffic.

Each node in the network maintains network topology based on both the HELLO and TC messages it receives. It then calculates and stores, for each node discovered, the shortest distance (i.e., the minimal required hops between the source and the destination) between itself and one of the destination's node MPRs; hence, the shortest path to the destination. See [4] for more details.

2.2 Node Isolation Attack

Kannhavong et al. proposed a Denial of Service (DOS) attack against OLSR called node isolation attack. In this attack, an attacker exploits the fact that the victim prefers a minimal MPR set in order to hide the existence of the victim in the network. The attacker, which must be located within broadcast distance of the victim, advertises a fake HELLO message claiming to be in close proximity to all of the victim's two-hop neighbors. In addition, a fictitious node is advertised, giving the attacker an advantage over other possible legitimate candidates for MPR selection. Knowledge of the victim's two-hop neighbors is readily available by analyzing TC messages of the victim's one-hop neighbors, a list of which can be constructed directly from the HELLO message broadcast by the victim himself. MPR selection rules would cause the victim to exclusively select the attacker as its sole MPR, as it is the minimal set that allows for coverage of all of the victim's two-hop neighbors (including the fictitious node).

DOS is now straightforward. The attacker can isolate the victim simply by not including the victim in its TC message. In essence, the attacker refrains from notifying the network that the victim can be reached through it, and because no other node advertises a path to the victim, it is isolated. Other nodes, not seeing link information to the victim, would conclude that it has left the network, and remove its address from their routing tables. Although nodes one- and two-hops from the victim would continue to exchange information with it, they will not propagate that information further as they were not designated as its MPR.

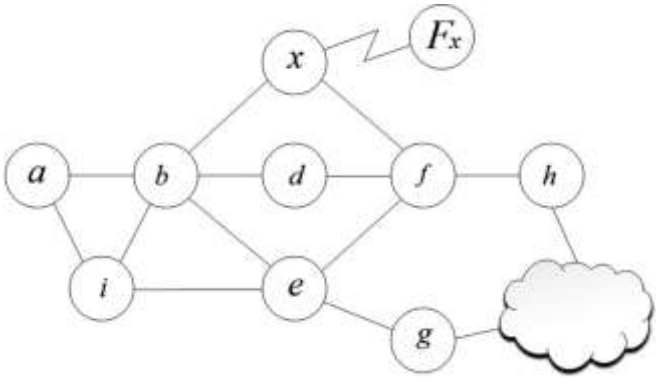


Fig. 1. Example of a node isolation attack: node x claims to know every two-hop neighbor of b, as well as Fx, a non-existent node.

The node isolation attack is illustrated by Fig. 1. Assume all nodes within broadcast distance have an edge connecting them, that node x is the attacker, that Fx is a fictitious node, and that node b is the victim. The cloud in the figure represents the rest of the network. OLSR rules state that x should have advertised a legitimate HELLO message containing {b, f}. Instead, it sends a fake HELLO message that contains {b, f, g, Fx}. This list contains all of b's two-hop neighbors, as well as one non-existent node, Fx. b would now innocently select x as its sole MPR, setting the ground for node isolation. By not advertising b in its TC message, x effectively isolates b from the rest of the network.

2.3 Related Work

There have been a number of solutions proposed in the literature for mitigating node isolation attack. The authors propose that every node inspect its MPRs' TC messages to see whether it has been included. This is possible due to the nature of the broadcast channel in wireless networks and also because MPR selection rules exclusively allow for the designation of MPRs within broadcast distance only. In Fig. 1, b can conclude whether x is malicious by looking for its own address in x's TC message; its lack thereof can only be due to malicious intent. This solution is elegant, but it has a number of drawbacks. First, this scheme is only effective against a single attacker, but, as the authors note, it fails in situations involving two consecutive colluding attackers. By having the first attacker orchestrates the attack yet advertise the correct TC, the victim cannot tell that it is under attack. The second colluding attacker, designated as the first's sole MPR, removes the victim from the advertised TC prior to propagation, isolating it from the network. This scheme is further limited; because detection only occurs after the attack has commenced,the scheme fails to preventin.

Kannhavong et al. attempt to mitigate the problem of colluding attackers. By modifying the HELLO message to include all two-hop neighbors, a node can detect existing contradictions between messages, thus identifying an attack. Of course, as the authors themselves noted, it is difficult to distinguish between contradictions which occur due to an attack as opposed to those resulting from topology changes. In addition, such contradictions identify an attack but fail to identify the culprit.

Raffo et al. propose a mechanism to improve the security of the OLSR routing protocol against external attackers. In their solution, each node signs its HELLO and TC messages. These signatures are later used by others to prove their own HELLO and TC messages. The resulting solution prevents devices from declaring imaginary links with known nodes. This solution functions correctly but is expensive in terms of overhead; besides the usual overhead of OLSR, signing messages requires extensive computation, a cumulative factor that grows as the size of the network increases. Another problem is the fact that the network loses its spontaneity as all nodes are required to know each other in advance in order to share their public keys. This prevents the network from evolving naturally from the various nodes that appear at a certain place and time, a fundamental trait of MANETs.

Another approach, based on local detection of link spoofing, is given by the authors provide a number of rules to identify abnormal behavior on the network. The solution includes a message sent in response to the detection of an intrusion, allowing for the exclusion of compromised nodes and preventing them from being included in network-wide routing tables. Besides the limited scope of the solution, as identification effectiveness is constrained to local nodes only, the ability of sending a warning message is disastrous in itself. Any malicious node can falsely advertise that some other node, local or remote, is malicious, causing for its immediate removal from routing tables all around. In a sense, the solution opens up an attack vector not present in the original problem.

Cryptographic primitives are used to secure communications (e.g., certificates, public key, and digital signatures), but they all require either a trusted third party or prior knowledge of network players.

More recently, use an internal reputation system in order to detect attacks. Distrust of nodes precludes them from being appointed as MPRs. In addition, they are able to identify a group that exhibits

malicious behavior. They are, however, unable to pinpoint the malicious node within the group and cannot tell, for example, which of the nodes in the path between the sender and receiver are colluding to execute the node isolation.

Dhillon et al. present an Intrusion Detection System (IDS) in which each node evaluates non-conformances of TCs with respect to previously known HELLO messages. This solution is effective under the assumption that HELLO messages can be trusted. In node isolation attack, however, the HELLO message itself is the problem. Indeed, the authors themselves mention the works as a methods for preventing spoofing attacks in HELLO messages. But, as we already mentioned, adds overhead to the network, as does by using control messages for verifying the HELLO messages.

A secure extension to the OLSR is proposed by Adjih et al.. A signature and timestamp is added to each control message. These enhancements prevent the modification and falsification of topology information and guarantee the timeliness of each message. This solution successfully blocks unauthorized users from joining an OLSR MANET, but cannot prevent attacks launched by compromised legitimate key-holding nodes.

Attempts to validate every node mentioned in the HELLO message a node receives. This is accomplished by adding two new control messages which are used for node verification. Upon receiving a new HELLO message, the would-be victim sends a two-hop verification request through pre-existing channels to every node claimed by the potential MPR (the attacker) to be its neighbor. In response, the queried nodes reply with their one-hop neighbor list. If the sender is present in all the reply messages, the node deduces that it's legitimate and can appoint it as MPR if it wishes. Otherwise, an attacker has been identified, and the presence of a malicious node is broadcast to the network. The attacker is subsequently removed from the routing tables throughout the network.

Besides lacking the elegance of previous solutions, this is plagued by a number of substantial problems. First, it is impossible to begin setting up a MANET as the first stages cannot be verified by any two-hop neighbors, because they don't exist yet. Second, a new node on the edge of the network legitimately recognized by a single node x would cause x to be incorrectly declared malicious. Nodes must meet a number of legitimate nodes before they can be safely introduced into the network. In addition, overhead is enormous as two or more messages are generated for each new HELLO message. In a mobile world where topology changes frequently, this enhancement is very costly. Finally, as in above, the option of declaring and broadcasting malicious nodes to the network opens up the possibility for a remote DOS, an attack vector not present in the original problem.

Another solution based upon the two-hop neighbors is described. Their solution, which is meant to deal with various link spoofing attacks, includes the addition of non-existent nodes. HELLO messages should be received from two-hop neighbors as well. A one-hop neighbor claiming knowledge about a new two-hop neighbor is not trusted until this information is verified either through a TC containing the two-hop node or a HELLO message emanating from that node. Ignoring the additional network overhead incurred, the solution is still lacking. A malicious node can falsify a new two-hop neighbor, and then corroborate this fallacy with a fake HELLO message. Alternatively, the attacker can broadcast a TC message claiming it has been nominated as MPR by the fictitious two-hop node.

Denial of Service Free OLSR (DFOLSR), which modifies the MPR selection process and adds two new control messages. Here too, corroboration messages are supplied by two-hop neighbors, with the node receiving the maximum number of replies selected as MPR. The authors claim that by not relying on one-hop neighbors, DFOLSR avoids node isolation attacks. Empirical evaluation of DFOLSR's cost is not provided, and an attacker falsifying the responses of fictitious two-hop nodes can render the solution useless.

Prevention measures based on message signing and countermeasures imposed when an attack is detected is the approach taken. Each new node initially sends its signature, which is later used to validate its messages. When an attack is detected (spoofed messages are incorrectly signed), countermeasures are imposed to isolate malicious nodes and ensure it does not participate in routing operations. A mechanism is also enabled for sharing the information regarding malicious nodes.

This solution generally functions well, but does not handle the case when an attacker joins the network prior to the victim allowing the attacker to impersonate the victim by sending false signature initiation data. In addition, a fake node sending HELLO messages (with fake signature initiation data) cannot be detected.

Suresh et al. investigate collusion attack in OLSR based MANETs. They propose a method called Forced MPR switching (FMS-OLSR) which requires that a node having a single MPR periodically change its MPR selection; thus, eliminating the necessary pre-condition for node isolation attack. This method might cause a legitimate network to temporarily fragment and is further limited because mitigation can only occur after the attack has commenced.

Generalized Intrusion Detection & Prevention (GIDP) and Intrusion Detection & Adaptive Response mechanism (IDAR) are examples of using Intrusion Detection Systems for solving MANET attack vectors. A survey of IDS based solutions can be found.

3 DENIAL CONTRADICTIONS WITH FICTITIOUS NODE MECHANISM

The first requirement of the proposed method is that each node will only use information available to it, without relying on any centralized or local trusted authority. Our technique does not actively verify the HELLO message, rather it checks its integrity by searching for contradictions between the HELLO message and the known topology. We allow for lone MPR nominations ,provided that no contradictions are found. Even in the face of contradictions, an MPR can be nominated for all two-hop neighbours for which it is the sole access point. It cannot, however, be nominated as sole MPR for two-hop neighbours that can be reached through other paths. Following ,we assume that TC messages can no the spoofed.

We justify this assumption due to the fact that bogus TC messages do not preclude a legitimate (attacked) victim from transmitting a valid TC that contradicts the bogus one. In essence, by publishing a fraudulent TC, the attacker discloses that he is attacking; allowing others to take preventive measures. A fake HELLO message is a much more crippling attack, because it removes a victim from the network without its knowledge. Hence, DOS and network disruption due to fraudulent TC messages is outside the scope of this paper.

For the remainder of this work, we use the following notation:

- V denote the set of all nodes in the network,
- v, x ∈ V are the victim (as well as/or the receiver) and attacker nodes, respectively,
- F_x is a fictitious node advertised by x,
- ADJ(v) ⊂ *V* is the set of all one-hop neighbors of v,
- ADJ2 (v) ⊂ Vis the set of all two-hop neighbors of v,
- MPR (v) ⊆ ADJ(v) is the set of one-hop nodes of v who appointed v as their MPR, and
- MPR' (v) ⊆ ADJ(v) is the set of one-hop nodes who were selected by v as MPRs.

DOI: 10.18535/ijecs/v6i6.38

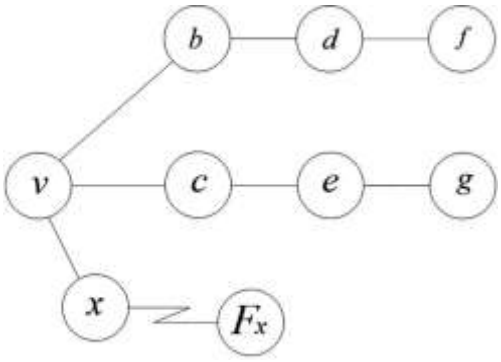


Fig. 2. Identifying contradictions to prevent node isolation attack.

3.1 Contradiction Rules

In this section we describe the rules that must be satisfied in order for a node to deem a HELLO message's sender trustworthy.

Consider Fig.2 where ADJ(v) = {b, c ,x}and ADJ2(v) = {d, e}. Based on OLSR, v must select MPR' (v) = {b, c} so that ADJ2(v) is covered. Suppose x is interested in isolating victim v. According to the attack presented in Section 2.2, x declares a fake HELLO message containing ADJ(x) = {v, d, e, F_x}.

x wouldn't declare {b, c} ⊆ ADJ(v), because v could verify this by comparing x's HELLO with the HELLO messages of b and c. Therefore, the first rule is:

1) When node x advertises a HELLO message containing ADJ(x), v should confirm that all of the nodes declared by x are not among ADJ(v).

   This can be accomplished by checking earlier HELLO messages to see whether or not they report the sender as their neighbor.

   As nodes b and c must exist in ADJ2(x), x must select MPRs that will allow it to reach these nodes. It might be the case, however, that x will pretend that it wants to choose v itself as MPR for covering b and c. Based on OLSR's, v cannot refuse. Under such a scenario, v cannot conclude that x is being malicious. However, v can check whether x appointed some other MPR for covering nodes in ADJ2(x) – {b, c}, namely either d or e. This brings us to the second rule:

2) For each node y mentioned in a HELLO message, v should examine whether there exists z ∈ADJ(y), such that (a) it is not mentioned in the sender's HELLO message and (b) is located at least three hops away from v. If these conditions are fulfilled, another examination is needed: (c) has x appointed w ∈ ADJ(x) as MPR for covering z?

For example, in Fig. 2, suppose x claims in its HELLO message that d is one of its neighbors. f is d's only neighbor located at least three hops away from v, while not being a one-hop neighbor of x. Based on x's message, f ∈ ADJ2(x); thus x should have appointed some node w ∈ ADJ(x) as MPR for covering f. This did not happen, thus causing a contradiction.

Examinations (a) and (b) could be done by searching within the TC table. If an entry containing the MPR that was appointed by x and allows it to reach z in only two hops does not exist, then there is a contradiction. Note that contradictions cannot be detected in cases in which either condition (a) or (b) is not fulfilled. In order to verify (c), v has to check each z ∈ Z, where Z ⊆ ADJ2(x) is based on x's message, and determine whether there is a TC message where either x has appointed z or z has appointed x as MPR.

A test of the condition (2a-c) is represented in Algorithm1, where the format of the TC message is {last (address), dest (address)}.

Algorithm 1. Testing-Condition-2

```
Testing-Condition-2(TC,G,x,v)
Z ← Ø
for each r ∈ TC do
  if r.last ∈ ADJ(x) do
    Z← Z ∪ {r.dest}
  if r.dest ∈ ADJ(x) do
    Z ← Z ∪ {r.last}
for each z ∈ Z do
  if z ∈ Z ∩ ADJ(v) do
    Z←Z - {z}
for each m 2 MPR' (x) do
for each z ∈ Z do
  if {m, x} 2 TC or {x; m} ∈ TC such that z is covered by m do
    Z← Z - {z}
  if Z ≠ Ø do
    mark x as suspected node
    else
    mark x as a legitimate MPR
```

A malicious node can try to evade the ramifications of the second rule by advertising that it is one hop from every node in V' where V'= V− ADJ(v). In order to prevent such an attack:

1) v must treat a HELLO message containing all ADJ(v) as an attack and take appropriate measures.

Nodes must apply each of the mentioned rules sequentially, advancing from one rule to the next iff there are no contradictions. If a contradiction is found, v should appoint x as a sole MPR only for the nodes that were exclusively declared in its HELLO message.

Although contradictions could temporarily exist, they should get resolved automatically during the link sensing process (in case of contradiction that arises from rule No. 1) or from the next TC messages (in case of contradiction that arises from rule No. 2) when they are received. Therefore, during the network initialization phase the number of MPRs may sometimes be higher than expected, but as the network grows and stabilizes their number settles into the normal range. We discuss this issue in Sections 4.2 and 4.3.

3.2 Using Fictitious Nodes

As described in the previous section, we only detect contradictions between a HELLO message and the network topology as is known to v based on proceeding HELLO and TC messages. We do not, however, verify every node that was mentioned in the HELLO message. As a result, there are scenarios where node isolation attack is still feasible. Consider, Fig. 3 in which x advertises that ADJ(x) = {v, e, c, g}, lying

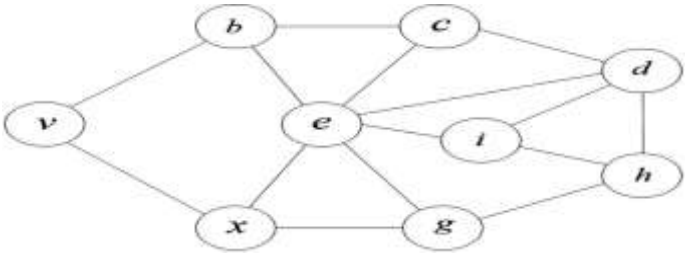


Fig. 3. An example of node isolation attack with no contradictions.

about the node c. ADJ2(x) = {b, c, d, i, h}, and MPR'(x) = {e, g}. v cannot identify any contradiction because:

- x doesn't claim to know any node, other than itself, contained in ADJ(v) (rule No. 1),
- x appointed MPRs for reaching all of ADJ2(x), namely, {b, c, d, i, h}. Thus, it is expected that x wouldn't appoint c as one of its MPRs, as d is already reachable by e (rule No. 2), and
- x doesn't claim to know all of ADJ(v), specifically {b} (rule No. 3).

This is an orchestrated scenario, but in a dense network it is easily feasible; thus, mitigation measures must be incorporated to avoid them.

Let us define a fictitious node, Fz, as a node declared by node z that doesn't actually exist. Fz is not declared fictitious, causing all other nodes believe it's a real node. This implies that all nodes will have an entry for Fz in their routing table, and all routes from or to Fz must pass through z. As a result, MPR(z) = ADJ(z), because each member of ADJ(z) has to appoint z as MPR.

Use of the fictitious node mechanism can prevent attacks similar to the case in Fig. 3 as well as others. If c declares a fictitious node, Fc, x has to appoint c as its MPR. Since this is something that can be verified as impossible through c's TC, x's lie can be caught. That is, the addition of fictitious nodes ensures that there will always be at least one node not included in the HELLO message so that x must appoint an MPR to cover it.

Unfortunately, if every node in the network were to declare an additional fictitious node, the network would revert to Link-State Routing, as all nodes would be nominated as MPRs due to their fictitious advertisement. Therefore, a mechanism for limiting fictitious announcements must be crafted, balancing between the need for node (and MPR) minimization and protecting the network from isolation attack.

3.3 Fictitious Setting Mechanism

In order to prevent nodes in the network from disseminating false information about their connectivity to the others, we set up a mechanism requiring each node to check whether an attack can be made through it. If such a lie is possible, the node adds a fictitious node to the network, preventing anyone from claiming false connectivity to this fake node. That is, responsibility for correctness of the connectivity information is delegated to the nodes themselves, as they must inhibit others from using them falsely. The limitation mechanism for adding or removing fictitious nodes is given by:

1) Each node v has to add a fictitious node when $\forall z \in ADJ2(v) \exists y \in ADJ(v)$ such that the distance between y and z<3-hops.
2) $Fv \notin ADJ(v)$.
3) New node z, advertises Fz by default, and only then calculates (1).
4) Removing the fictitious node is done when (1) is false.
5) Examination must be performed periodically (every FICTITIOUS_CHECK_INTERVAL 1).

Nodes $\{x, i\} \subset ADJ2(c)$ in Fig. 3 do not contain any node with a distance of three from each of them. Therefore, based on rule No. 1 of the fictitious setting mechanism, node c must add a fictitious node to the network. This counters the attack and protects node v, as node x must appoint c as an MPR in order to reach Fv. This will contradict rule No. 2 of the contradiction rules (Section 3.1) and will be flagged. Clearly, at no stage is the fictitious node denoted as fictitious. Only the advertiser v knows it is fictitious.

The authors claimed that "the larger and more dense a network, the more optimization can be achieved as compared to the classic link state algorithm." Similarly, our simulations show that as the number of the nodes in the network grows, the number of fictitious nodes required decreases. More details will be specified later, in Sections 4.2, 4.3.

4 C OST,SIMULATION MODEL, AND RESULTS

4.1 Cost Estimation Metrics

In order to compare the overall cost of DCFM, we must compare the additional overhead DCFM introduces as compared to OLSR without node isolation mitigation. Given that fictitious nodes don't send HELLO messages, as they are—after all—fictitious, DCFM only imposes additional overhead by influencing the size and number of TC messages in the network. The difference between the average size of TC messages when DCFM is active and the average size of TCs using standard OLSR on the same network, represents the average increase in TC size due to DCFM. Performing this calculation on the expected number of MPRs will provide the expected increase in the number of TCs transmitted.

Equation 1 gives the average size of TCs for a given topology. This weighted mean is equal to number of the nodes who selected v as their MPR divided by the total number of nodes in the network

Using the notation from 3 and letting $n = |V|$ be the size of the network, we get

$$\frac{1}{n}\sum_{v=1}^{n}\sum_{v\prime\in MPR(v)} 1 \qquad (1)$$

In equation 2 we calculate the probability that a randomly selected node is an MPR:

$$\frac{1}{n}\sum_{v=1}^{n}\frac{\sum_{v\prime\in MPR} 1}{|ADJ(v)|} \qquad (2)$$

Thus, when all nodes select all of their neighbors as MPRs, such as a topology looking like a long chain of nodes, OLSR reduces to LSR and rand(x) = 1 as expected. Of course, we assume that $|ADJ(v)| \neq 0$, as a node without any neighbors, isn't part of the network. In equation 3 we calculate the probability that a randomly selected node is mistakenly suspected as malicious, causing v not to appoint it as an MPR. Allowing $SUSPECT(V) \subseteq ADJ(v)$ to be the set of one-hop nodes falsely suspected by v,

Percent of the wrongly suspected nodes in the network =

$$\frac{1}{n}\sum_{v=1}^{n}\frac{\sum_{v\prime\in SUSPECT(v)} 1}{|ADJ(v)|} \qquad (3)$$

4.2 Simulation

The cost metrics of Section 4.1 are only accurate for a given topology. As network topologies are infinite, expected cost estimation must be achieved through simulation. We used the built-in OLSR module in the network simulator NS3. It was augmented to run DCFM in accordance with the protocol above. All simulation value sets were run ~1,000 times, with values reported as averages over these results. The movement, where relevant, was 1.5-2 m/s (5.4-7.2 km/h).

The first set of simulations was designed to test the effectiveness of DCFM against node isolation attacks. For this purpose we ran three

DOI: 10.18535/ijecs/v6i6.38

different simulations: without movement, with movement using a single attacker, and with movement when a colluding attack is in progress. Each of the first two simulations used 30 nodes in random topology in an area of 750 1,000 m. In addition, three predefined nodes were used: the victim, the attacker, and a sender used for sending messages to the victim. Both the victim and the sender were randomly placed at a distance of at least three hops from each other. The reason for this restriction is because according to the OLSR RFC, packets sent from a distance of one or two hops do not use the TC table and are thus not affected during the isolation attack. The attacker, however, was designed to follow the victim. The transmission range was about 250 meters.

In the colluding attack simulation, 38 nodes were used, following, eight of which were predefined attackers located stationed at equal distances from each other so that every point in the area was covered by at least one attacker. The other 30 nodes were allowed to move freely. All other constraints were similar to that which was described above for the other simulations.

In each simulation we calculated, based on equations 1 and 2, the percentage of messages received by the victim out of the total number of messages sent. Each simulation round was ran using the same topology and seed:

TABLE 1
The Percentage of Received Messages with and without Attack, with DCFM Turned On and Off

Attack: DCFM:	False off	False on	True off	True on
Movement: false	86.94	86.72	0	86.9
Movement: true	69.11	69.07	24.58	68.03
Colluding & Movement	84.14	84.52	24.89	79.45

without attack, under attack, with and without the protection of DCFM.

A second set of simulations was done to explore the overhead costs as the number of nodes in the network grows, as well as to find out the average size of a TC message. In this simulation set, all values were constant except for a random network topology with a varying number of nodes, fluctuating network density from 50 to 400.

The last simulation was designed to evaluate the false positive ratio, namely, the number of legitimate nodes which were wrongly suspected as defined in equation 3. Again, all values were kept constant except for the network topology which was random. For these last two simulations there was no movement in the network. This was done in order to isolate simulation values and avoid possible interference. For example, the overhead measurement naturally increases when movement in the network is present irrespective of DCFM.

4.3 Results and Discussion

In this section we show the results of the simulations mentioned in Section 4.2. Each row in Table 1 represents a simulation designed to test the effectiveness of DCFM. Each simulation was tested with and without attack, while DCFM protection is turned on and off. The percentages in the table are the average percentage of messages received by the victim at every stage. When there is no movement, fewer messages are naturally lost; increasing the success metrics. This is the reason why in the third simulation, when no attack was carried out, the results are substantially better than the same simulation under attack with the protection of DCFM. It also explains why some of the messages get through even though the network is under attack and no protection is applied.

According to the third simulation settings, attacking nodes are fixed without any movement. This explains why the values of the third row, both when attack is executed (column 3) and when it isn't (column 1) are better than the values in the second row (all nodes are moving). The discrepancy between when under attack and when attack isn't executed can be attributed to a real bottleneck in the network; a situation that is independent from the attack and defense. Thus, even with the protection of DCFM, the results are slightly worse than the same scenario without attack.

Fig. 4 depicts the number of fictitious nodes required as a function of node density. The number of fictitious nodes was estimated using the fictitious setting mechanism. While the X-axis represents the number of nodes in a random network topology, the Y-axis represents the average percentage

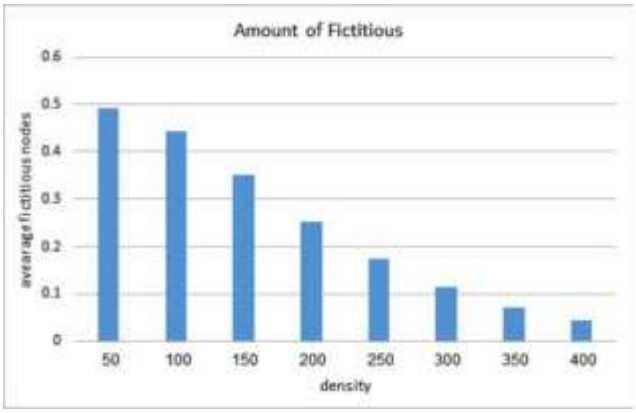


Fig. 4. Number of required fictitious nodes, depending on the network density.

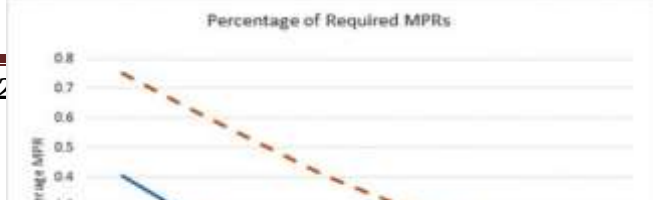
of fictitious nodes as a function of the number of nodes that were added to the network (the actual overhead).

Fig. 5 presents the overhead costs as the number of nodes in the network grows. The X-axis represents the number of nodes in a random network topology, while the Y-axis represents the percentage of nodes, again, as a function of nodes in the network that were selected as MPRs. R.MPR represents the number of required MPRs in regular OLSR. The second curve represents the number of required MPRs when DCFM is active and is denoted by DCFM.MPR. Both are displayed as percentages of the total population size.

In Fig. 6, R.TC represents the average size of a TC message in regular OLSR, DCFM.TC represents the average size of a TC message when DCFM is active, and LSR.TC is the average size of a TC message in Link-State Routing protocol. All the three measures are displayed as percentages of the total population size.Last, our experimentation with 1,000 random topologies show that the average number of nodes mistakenly suspected as malicious stands at less than 5 percent.

To conclude, our simulations show that

1) DCFM effectively defends OLSR from node isolation attack, even when every node in the network is allowed to move.



DOI: 10.18535/ijecs/v6i6.38

Fig. 5. Number of required MPRs, depending on the network density.

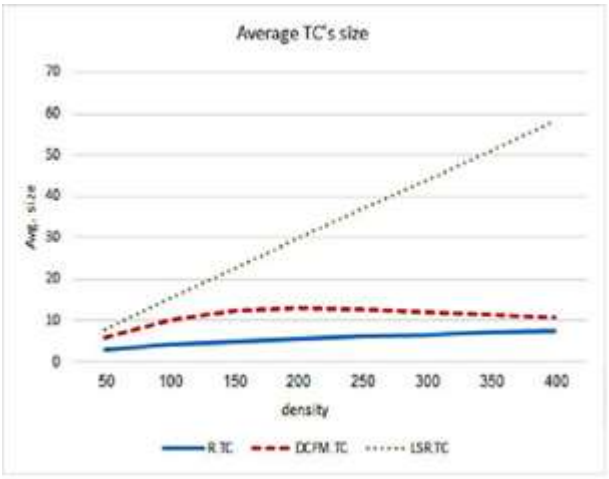


Fig. 6. The average size of a TC message, depending on the network density.

2) Basically, there are two approaches to protecting adhoc networks: prevention-based approaches and detection-based approaches. DCFM belongs to the first class since it prevents the attack by not appointing the attacker as MPR.
3) Fig. 5 clearly shows that as the population grows, the percent of overhead decreases. Moreover, as the population density increases, the difference between DCFM.MPR and R.MPR decreases.
4) Since each node in DCFM prevents the attack personally, additional attacks resulting from the solution (such as advertising a possible malicious node to the network), are impossible.

An additional comparison simulation was conducted to compare the solution. As the authors themselves noted, their solution worked well when the attack was carried out by a single node and completely failed by using a colluding attack. In their solution, 90 percent of the messages that were sent to the node were received under the first attack but 0 percent were received under the collusion attack. In our solution, however, about 87 percent of the messages were received in the first attack and about 80 percent under the collusion attack (see Table 1).

False negative cases. Although DCFM prevents the vast majority of node isolation attacks in the network as reflected by our simulation, there are rare cases in which the attacker can overcome DCFM. However, such cases are unusual, so it is extremely difficult for the attacker to plan and execute such attacks.

An example of this can be seen in the following scenario (see Fig. 7) in which every node ai has a fictitious node, and the cloud on the right represents the rest of the network.

As usual, x is interested in isolating v. x advertises a HELLO message containing ADJ(x) = {v,d,Fc,F_a},where F_a represents the fictitious nodes of every ai. There is no contradiction to the rules mentioned in Section 3.1: (1) there is no contradiction between the HELLO message of x and ADJ(v), (2) x covers every ADJ2(v) node and there is no node z three hops from v that would've required x to nominate an MPR from ADJ2(v), and (3) x's message does not mention all the network nodes.

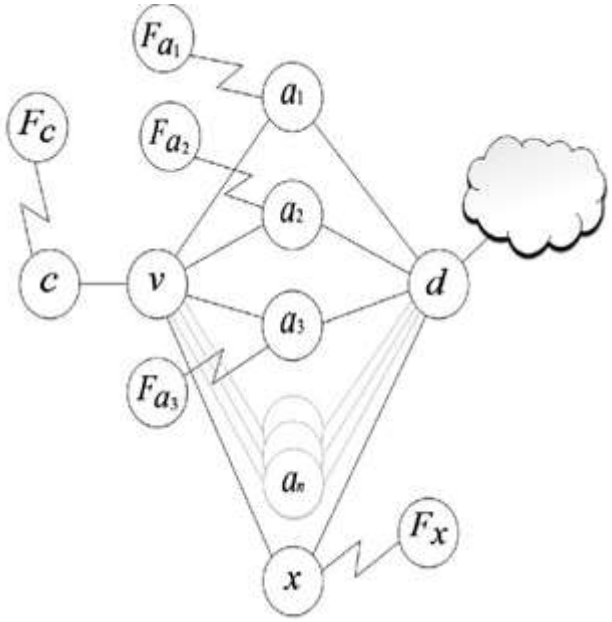
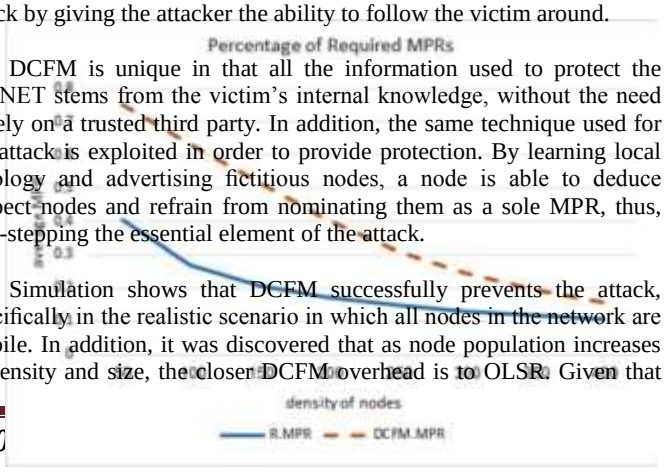


Fig. 7. An example of the case of a false negative attack

In this case, since all of x's lies are related to fictitious nodes, everyone—except for v—would know that x is malicious. Such cases can occur when connectivity to the rest of the network is limited. This, however, isn't within x's control; rather, it is a random topology that x can take advantage of. Moreover, the results of the simulation without movement (see the first row of Table 1) indicate that such cases are extremely rare, because the number of successful messages received while under the protection of DCFM is equal to the number of received messages when no attack is executed.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a solution called DCFM whose function is to prevent a node isolation attack in which the attacker manipulates the victim into appointing the attacker as a sole MPR, giving the attacker control over the communication channel. We further strengthened the attack by giving the attacker the ability to follow the victim around.



DCFM is unique in that all the information used to protect the MANET stems from the victim's internal knowledge, without the need to rely on a trusted third party. In addition, the same technique used for the attack is exploited in order to provide protection. By learning local topology and advertising fictitious nodes, a node is able to deduce suspect nodes and refrain from nominating them as a sole MPR, thus, side-stepping the essential element of the attack.

Simulation shows that DCFM successfully prevents the attack, specifically in the realistic scenario in which all nodes in the network are mobile. In addition, it was discovered that as node population increases in density and size, the closer DCFM overhead is to OLSR. Given that

OLSR functions best in dense large networks, DCFM can function without real additional cost.

We expect that with only minor adjustments, DCFM can protect OLSR from the family of attacks that centres around the falsification of HELLO messages with the intention of being appointed as sole MPR (e.g., black hole, gray hole, and

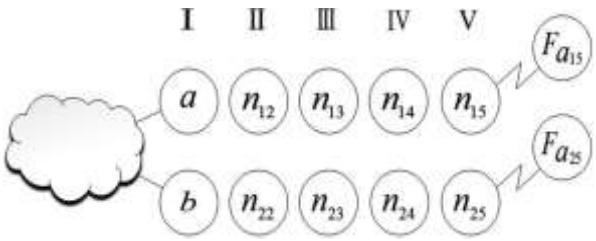


Fig. 8. An example to a mistakenly suspicion in a legitimate node.

wormhole attacks). We leave this for future work. We also leave to further research the exact values of FICTITIOUS_CHECK_INTERVAL that minimize the overall computation but still leave mitigation active and responsive.

REFERENCES

[1] S. Mclaughlin, D. Laurenson, and Y. Tan, "Mobile ad-hoc network." (Aug. 10 2006) uS Patent App. 11/351,777. [Online]. Available: http://www.google.com/patents/US20060176829

[2] C. E. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance-vector routing (dsdv) for mobile computers," in Proc. Conf. Commun. Archit., Protocols Appl., 1994, pp. 234–244.

[3] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in Proc. IEEE Int. Multi Topic Conf. Technol., 2001, pp. 62–68.

[4] T. Clausen and P. Jacquet, "RFC 3626-Optimized Link State Routing Protocol (OLSR)," p. 75, 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

[5] D. Johnson, Y. Hu, and D. Maltz, "Rfc: 4728," Dynamic Source Routing Protocol (DSR) Mobile Ad Hoc Netw. IPV4, 2007. [Online]. Available: http://tools.ietf.org/html/rfc4728

[6] C. Perkins and E. Royer "Ad-hoc on-demand distance vector routing," in Proc. 2nd IEEE Workshop Mobile Comput. Syst. Appl., Feb. 1999, pp. 90–100.

[7] E. Gerhards-Padilla, N. Aschenbruck, P. Martini, M. Jahnke, and J. Tolle, "Detecting black hole attacks in tactical manets using topology graphs," in Proc. 32nd IEEE Conf. Local Comput. Netw., Oct. 2007, pp. 1043–1052.

[8] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, and D. Raffo, "Securing the olsr protocol," in Proc. Med-Hoc-Net, 2003, pp. 25–27.

[9] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, and A. Jamalipour, "A survey of routing attacks in mobile ad hoc networks," IEEE Wireless Commun., vol. 14, no. 5, pp. 85–91, Oct. 2007.

[10] D. Dhillon, J. Zhu, J. Richards, and T. Randhawa, "Implementation & evaluation of an ids to safeguard olsr integrity in manets," in Proc. Int. Conf. Wireless Commun. Mobile Comput., 2006, pp. 45–50.

[11] D. Raffo, C. Adjih, T. Clausen, and P. M€uhlethaler, "An advanced signature system for OLSR," in Proc. 2nd ACM Workshop Security Ad Hoc Sensor Netw., 2004, pp. 10–16.

[12] C. Adjih, D. Raffo, and P. M€uhlethaler, "Attacks against OLSR: Distributed key management for security," in Proc. 2nd OLSR Interop/Workshop, Palaiseau, France, 2005.

[13] Y.-C. Hu, A. Perrig, and D. Johnson, "Wormhole attacks in wireless networks," IEEE J. Selected Areas Commun.., vol. 24, no. 2, pp. 370–380, Feb. 2006.

[14] B. Kannhavong, H. Nakayama, and A. Jamalipour, "Nis01-2: A collusion attack against olsr-based mobile ad hoc networks" in Proc. IEEE Global Telecommun. Conf., Nov. 2006, pp. 1–5.

[15] B. Kannhavong, H. Nakayama, N. Kato, Y. Nemoto, and A. Jamalipour, "Analysis of the node isolation attack against olsr based mobile ad hoc networks," in Proc. Int. Symp. Comput. Netw., 2006, pp. 30–35.

[16] M. Wang, L. Lamont, P. Mason, and M. Gorlatova, "An effective intrusion detection approach for olsr manet protocol," in Proc. 1st IEEE ICNP Workshop Secure Netw. Protocols, Nov. 2005, pp. 55–60.

[17] D. Dhillon, T. Randhawa, M. Wang, and L. Lamont, "Implementing a fully distributed certificate authority in an olsr manet," in Proc. IEEE Wireless Commun. Netw. Conf., Mar. 2004, vol. 2, pp. 682–688.

[18] M. Omar, Y. Challal, and A. Bouabdallah, (2009). "Reliable and fully distributed trust model for mobile ad hoc networks.," Comput. Security [Online]. vol. 28, pp. 199–214. Available: http://www.sciencedirect.com/science/article/pii/ S016740480800117X

[19] A. Adnane, C. Bidan, and R. T. de Sousa Jnior, "Trust-based security for the olsr routing protocol," Comput. Commun., vol. 36, no. 10, pp. 1159–1171, 2013.

[20] F. Hong, L. Hong, and C. Fu, "Secure olsr," in Proc. 19th Int. Conf. Adv. Inform. Netw. Appl., Mar. 2005, vol. 1, pp. 713–718.

[21] M. Marimuthu and I. Krishnamurthi, "Enhanced olsr for defense against dos attack in ad hoc networks," Commun. Netw., J., vol. 15, no. 1, pp. 31–37, Feb. 2013.

[22] Y. Jeon, T.-H. Kim, Y. Kim, and J. Kim, "Lt-olsr: Attack-tolerant olsr against link spoofing," in Proc. IEEE 37th Conf. Local Comput. Netw., 2012, pp. 216–219.

[23] D. Malik, K. Mahajan, and M. Rizvi, "Security for node isolation attack on olsr by modifying mpr selection process," in Proc. 1st Int. Conf. Netw. Soft Comput., Aug. 2014.

[24] A. Adnane, C. Bidan, and R. de Sousa, "Trust-based countermeasures for securing olsr protocol," in Proc. Int. conf. Comput. Sci. Eng., Aug. 2009, vol. 2, pp. 745–752. [25] P. Suresh, R. Kaur, M. Gaur, and V. Laxmi, "Collusion attack resistance through forced mpr switching in olsr," in Proc. Wireless Days, Oct. 2010, pp. 1–5.

[26] A. Nadeem and M. Howarth, "Protection of manets from a range of attacks using an intrusion detection and prevention system," Telecommun. Syst., vol. 52, no. 4, pp. 2047–2058, 2013.

[27] A. Nadeem and M. P. Howarth, (2014). An intrusion detection & adaptive response mechanism for manets. Ad Hoc Netw. [Online]. vol. 13, pp. 368–380. Available: http://www.sciencedirect.com/science/article/pii/S1570870513001959

[28] A. Nadeem and M. Howarth, "A survey of manet intrusion detection & prevention approaches for network layer attacks," IEEE Commun. Surveys Tuts., vol. 15, no. 4, pp. 2027–2045, Oct.-Dec. 2013.

[29] D. Raffo, "Security schemes for the olsr protocol for ad hoc networks," Ph.D. thesis, Universite Paris, 2005.

[30] [Online]. Available: http://www.nsnam.org/, Oct. 2013.

[31] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in mobile ad hoc networks: Challenges and solutions," IEEE Wireless Commun., vol. 11, no. 1, pp. 38–47, Feb. 2004.

[32] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on demand routing protocol for ad hoc networks," Wireless Netw., vol. 11, no. 1-2, pp. 21–38, Jan. 2005.