Misbehaviors Detection to Ensure Availability in OLSR

Frédéric Cuppens, Nora Cuppens-Boulahia, Tony Ramard, Julien Thomas

GET/ENST Bretagne, 2 rue de la Châtaigneraie, 35512 Cesson Sévigné Cedex, France

Abstract. In this paper, we investigate the use of Aspect-Oriented Programming (AOP) [13] in the domain of Mobile Ad-hoc NETworks (MANETs). More precisely we study the availability issues in Proactive routing protocols. This paper classifies the different possible attacks and examines the countermeasures to ensure availability. Our approach is based on a detection-reaction process. The reasoning followed by the detection process is built on a formal description of normal and incorrect node behaviors. This model allows us to derive security properties. These properties are woven into our implementation using the AOP. Our algorithm checks if these security properties are violated. If they are, detection of incorrect (malicious) behaviors occurs to allow the normal node to find a path without incorrect node behavior. Therefore the detector node sends to its neighborhood the detection information to allow its neighbors to avoid choosing the intruder as a node to cross to. A node chooses the path using its local diagnosis and the reputation of other nodes. Using a field in the standard control message to communicate the detections, our approach does not change the message format, so it is very easy to use and there is no overhead. While we use OLSR as an example of protocol for our studies, we argue that the presented techniques apply equally to any proactive routing protocol for MANETs.

Key words: Mobile Ad Hoc Network, Intrusion Detection, Availability, OLSR, Routing.

1 Introduction

A Mobile Ad-hoc NETwork (MANET) is a collection of nodes which are able to connect to a wireless medium forming an arbitrary and dynamic network. The routing protocol ensures that all nodes at all times can reach all destinations in the network. However several attacks can occur against security in order to disrupt the network.

In this paper, we investigate the issues of intrusion detection and response in MANET. As a main result, we provide a security extension to OLSR, a proactive MANET routing protocol. Our primary issue with respect to securing MANET routing protocols is to ensure the network integrity, even in presence of malicious nodes. It is not our propose in this paper to deal with node authentication which is an issue already investigated elsewhere[14]. Our approach is based on a formal security model called Nomad [7]. This model allows us to express node behaviors (normal and incorrect behaviors). From these expressions, we can derive properties to specify a security policy. These properties are woven into the routing protocol using an Aspect-Oriented Programming (AOP). These properties are checked when a message is received in order to detect intrusions. If a property is violated, a reaction occurs and the node attempts to find another path or Multipoint Relay (MPR) keeping the malicious node away. In this case, the node sends relevant information related to the detection to its neighborhood. The neighbors of this node record this information but do not fully trust it. A function allows nodes to compute the reputation in their neighbors. The reputation quantification allows nodes to choose the best path to reach another node.

The remainder of this paper is organized as follows. Section 2 presents the different kinds of Ad hoc routing protocol especially OLSR. Section 3 describes the vulnerabilities of Ad hoc routing protocols including OLSR. In section 4 we present related works. Section 5 gives an outline of our approach to satisfy availability requirements in ad hoc networks and briefly presents the modeling language we choose to express these availability properties and to specify node profiles. In Section 6, we define the node profiles and availability properties to detect and to communicate malicious behaviors and we show how these properties are woven with AOP into the code to secure the OLSR protocol. Section 7 is an experimentation of our mechanism to secure OLSR based on these properties and section 8 concludes.

2 Mobile Ad hoc Network (MANET)

In Ad hoc networks, to ensure the delivery of a packet to a destination node, each node must run a routing protocol and maintain its routing tables in memory. Routing protocols can be classified into the following categories: reactive, proactive, and hybrid.

In this section, we present the Optimized Link State Routing protocol (OLSR) [4] using as an example to illustrate our approach. OLSR is a proactive routing protocol, designed specifically for large and dense MANETs. It is based on a Multipoint Relaying (MPR) flooding technique to reduce the number of topology broadcast packets, see figure 1.

2.1 Overview

Every node broadcasts HELLO messages that contain one-hop neighbor information periodically. If the Time To Live (TTL) of HELLO message is 1, the message is not forwarded. With the aid of HELLO messages, every node obtains local topology information.

A node (also called selector) chooses a subset of its neighbors to act as its Multipoint Relaying (MPR) nodes. This choice is based on the local topology information carried by HELLO messages. MPR nodes have two roles:

- When the selector sends or forwards a broadcast packet, only its MPR nodes among all its neighbors forward the packet;
- The MPR nodes periodically broadcast its selector list throughout the MANET in TC (Topology Control) message. Thus every node in the network knows by which MPR node the target node could be reached.

Notice that there is no guarantee that the selected MPR node is not a malicious node.

With global topology information stored and updated at every node, a shortest path from one node to every other node could be computed with Dijkstra's algorithm [8], which goes along a series of MPR node.

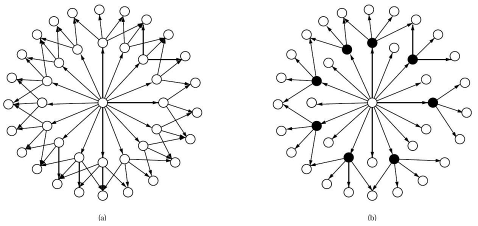

Fig. 1. Two hop neighbors and 'multipoint relays' (the solid circles) of a node. (a) illustrates the situation where all neighbors retransmit a broadcast, (b) illustrates where only the MPRs of a node retransmit the broadcast[4]

3 Security Flaws

In this section, we discuss various security vulnerabilities in ad hoc network.

One vulnerability, common to all routing protocols operating a wireless adhoc network, is 'jamming', i.e. a node generates massive amounts of interfering radio transmissions. In this paper, we do not consider a network resistance against jamming nor traffic overloading.

Attacks against MANETs can be divided into two groups: Passive attacks typically involve only eavesdropping of data whereas active attacks involve actions performed by adversaries, for instance the replication, modification and deletion of exchanged data. External attacks are typically active attacks that are targeted to prevent services from working properly or shut them down completely.

In summary, a malicious node can disrupt the routing mechanism employed by several routing protocols in the following ways. It attacks the route discovery process by generating link spoofing or identity spoofing, changing the contents of a discovered route, modifying a route reply message, causing the packet to be dropped as an invalid packet, invalidating the route cache in other nodes by advertising incorrect paths, refusing to participate in the route discovery process. The malicious node attacks the routing mechanism by modifying the contents of a data packet or the route via which that data packet is supposed to travel, behaving normally during the route discovery process but dropping data packets causing a loss in throughput.

These vulnerabilities makes it clear that ad hoc networks are inherently insecure, more so than their wireline counterparts, and need a mechanism to counter attacks on a system as soon as possible (ideally in real time) and take appropriate action.

4 Related Works

Sergio Marti et al. discussed two techniques that improve throughput in MANETs in the presence of compromised nodes that agree to forward packets but fail to do so [15]. A node may misbehave because it is overloaded, selfish, or broken. However, a node can only detect these behaviors and does not communicate this detection to its neighborhood. We take into account this approach, and we add the way to communicate the detection information to the neighborhood.

Bhargava et al. [3] proposed an intrusion detection and response model (IDRM) to enhance security in the Ad Hoc On Demand Distance Vector (AODV) routing protocol. When the misbehavior count for a node exceeds a predefined threshold, the information is sent out to other nodes as part of global response. However in this approach, a collusion of nodes can eject a normal node from the network. In our approach we adopt the communication of the detection information. A node sends periodically to its neighborhood the trust level of each neighbor.

In [2] the authors propose to associate one signature with each OLSR message, rather than one for each OLSR packet. In addition to this, one timestamp is provided for each signature. The timestamps are used to assess the freshness of the messages, thus avoiding replay attacks. The signature is encapsulated and transmitted as an ordinary OLSR message. This means that the signature and the message can travel in separate packets and separate routes from the originator. Also, the proposed system in [2] is an end-to-end system. The suggested timestamp exchange protocol proposed in [2] is a rather complex solution.

Most of the research works (like [12] [11]), attempt to apply cryptography techniques to secure MANET routing protocols. But, we know, as in wired network, that in addition to intentional and not intentional malicious behaviors there are always design flaws, human errors that enable attackers to exploit software vulnerability. Hence, we follow a property oriented intrusion detection approach and develop a reaction mechanism to deal with the detected intrusions. The difference between our approach and these related works is the fact that we do not change the message format. Consequently our algorithm is easier to implement. In section 5 we explain the model we use to define the node profiles and the properties woven in our AOP approach. These properties are the orthogonal aspect in our approach.

5 Modeling Approaches

To study different availability properties in mobile ad-hoc networks (MANET), we take into account topological dimension. This study is based on the properties of topological information exchanged during the network building and the topological maintenance. The regular network maintenance between nodes allows the discovery of the available routes and the participant nodes. Each node provided with a sensor, analyzes and detects errors in “control messages” exchanged between the nodes and readjust, if possible, its routing tables in compliance with this analysis.

When dealing with security properties like availability, classical first order logics are no longer appropriate. We need a more expressive security model such as the Nomad model [7]. Nomad is a security model based on deontic logic and temporal and temporized logics of actions which provides expressiveness necessary to specify availability requirements. Thanks to the Nomad model, we specified the OLSR protocol and expressed availability properties. In this paper, we only use the temporal and temporized framework of Nomad. Thus, we introduce the temporal modality $\Box A$ and the temporized modality $\bigcirc^{\leq d}A$, for $d \geq 0$. If A is a formula, then $\Box A$ is to be read “A is always true” and $\bigcirc^{\leq d}A$ is to be read “A is eventually true within a delay of d units of time”. Using these modalities, we can express two availability properties:

- ”Usual” availability: a message m must be received by a node nd in maximum delay d each time it is sent by a node ns

$$(a) \qquad \Box(SEND(ns, nd, m) \rightarrow \bigcirc^{\leq d}RECEIVE(ns, nd, m))$$

- Weak availability: a message m must be received by a node nd in maximum delay d each time it is sent by a node ns and there exists a route, a transitive closure of symmetric links, between ns and nd.

$$\begin{aligned}(b) \qquad \Box(&SEND(ns, nd, m) \wedge ROUTE(ns, nd)\\ &\rightarrow \bigcirc^{\leq d}RECEIVE(ns, nd, m))\end{aligned}$$

We try to satisfy the property (b), as it is the most compliant availability property with the characteristics of ad-hoc networks. For this purpose, we shall derive more basic properties (see the following section) from the protocol specification.

6 OLSR Availability Analysis

Each node has a view of the network topology derived from (Hello and TC) messages it receives. This view can be modified by a malicious node and an attack against the availability can occur. To study different availability properties, node profiles have to be specified (6.1) to understand the behavior of normal and malicious nodes. Thanks to theses profiles and the messages, basic properties (6.2) can be expressed and have to be satisfied to ensure availability.

6.1 Node Profiles

MANET nodes which participate in the network routing can be grouped by the way they act in the network. The identified behaviors of nodes are called node profiles. These profiles are very important to understand how a normal node and an intruder work. Thus, we can derive properties to identify theses profiles.

- Profile of a cooperative MPR node nb who always transmits TC messages when it receives them from its neighbor na before the expiration of time Max.

$$\begin{aligned}\boldsymbol{COOPERATIVE(nb)} \leftrightarrow \Box(RECEIVE(na, nb, m) \wedge \\ NEIGHBOR(nb, nv) \wedge MPR\_NEIGHBOR(na, nb) \\ \rightarrow \bigcirc^{\leq Max} PROPAGATE(nb, nv, m))\end{aligned} \tag{1}$$

- Profile of a "lazy node" nb who transmits messages irregularly.

$$\boldsymbol{LAZY(nb)} \leftrightarrow \neg COOPERATIVE(nb) \tag{2}$$

- Profile of an "selfish node" nb who never transmits messages. An egoist MPR node receives TC messages directly from the sender or other MPR relays but it does not transmit those messages.

$$\begin{aligned}\boldsymbol{SELFISH(nb)} \leftrightarrow \Box(RECEIVE(na, nb, m) \wedge \\ MPR\_NEIGHBOR(na, nb) \wedge NEIGHBOR(nb, nv)) \\ \rightarrow \Box \neg PROPAGATE(nb, nv, m)\end{aligned} \tag{3}$$

- Profile of a "slanderer node" that generates incorrect information. Such a node can forward incorrect information (carried by control messages) received from other nodes.

$$\begin{aligned}&\boldsymbol{SLANDERER(nb)} \leftrightarrow \\ &(\neg TC(na, nb, m) \wedge NEIGHBOR(nb, nv) \wedge \\ &MPR\_NEIGHBOR(na, nb) \wedge PROPAGATE(nb, nv, m)) \\ &\vee \\ &(\neg HELLO(na, nb, m) \wedge NEIGHBOR(na, nb) \wedge \\ &SEND(nb, na, m))\end{aligned}$$

- Profile of a "secretive node" (malicious MPR node) that does not forward any correct message which has to be forwarded through this node.

$$\begin{aligned}&\boldsymbol{SECRETIVE(nb)} \leftrightarrow \\ &(TC(na, nb, m) \wedge BELIEVE(nb, m) \wedge \\ &MPR\_NEIGHBOR(na, nb) \wedge NEIGHBOR(nb, nv) \wedge \\ &\neg PROPAGATE(nb, nv, m)) \\ &\vee \\ &(HELLO(na, nb, m) \wedge BELIEVE(nb, m) \wedge \\ &NEIGHBOR(na, nb) \wedge \neg SEND(nb, na, m))\end{aligned}$$

- Profile of a "liar node" nb that generates incorrect information or does not forward any correct message which has to be forwarded:

$$\boldsymbol{LIAR(nb)} \leftrightarrow SLANDERER(nb) \vee SECRETIVE(nb) \tag{4}$$

- Profile of a "honest node" nb that sends only correct routing information:

$$\boldsymbol{HONEST(nb)} \leftrightarrow \Box\neg LIAR(nb) \tag{5}$$

Among these profiles we are particularly interested in the profile of liar node that we use to derive the properties shown in the section 6.2.

6.2 Basic Properties Specification For Detection Of Liar Nodes

Using the characteristics of OLSR (section 2), we can derive some properties that allow us to detect the inconsistencies in OLSR control messages.

Hello-TC Relationship Property: For a MPR node, all its MPR selectors carried by TC messages are always found among all the one-hop neighbors carried by Hello message.

$$\begin{aligned} HELLO(na, nb, m) \wedge TC(na, nb, m') \wedge NEXT(m, m') \\ \wedge MPR(m', n_c) \rightarrow NEIGHBOR(m, n_c) \end{aligned} \tag{6}$$

MPR-MPR Selector Relationship Property:

- When a node nb receives a TC message from node na, if node nb is claimed as node na's MPR selector, then node nb must have chosen node na as its MPR.

$$\begin{aligned} TC(na, nb, m) \wedge IN\_MPRS\_SET(nb, m) \\ \rightarrow MPR\_NEIGHBOR(nb, na) \end{aligned} \tag{7}$$

- When a node nb receives a TC message from node na, if another node nc is claimed as MPR selector of node na, then node nc must have chosen node na as its MPR and declared that in its previous Hello message.

$$\begin{aligned} TC(na, nb, m) \wedge HELLO(nc, nb, m') \wedge \\ IN\_MPRS\_SET(nc, m) \rightarrow IN\_MPR\_SET(na, m') \end{aligned} \tag{8}$$

Message Integrity Property: When a MPR node $n1$ receives a TC message and if this message must be forwarded via node $n1$, then the TC message must not be modified by node $n1$. The same copies of the TC message must be received by its originator and all MPR nodes who have forwarded this TC message.

$$\begin{aligned} TC(ns, n1, m) \wedge MPR\_NEIGHBOR(ns, n1) \\ \rightarrow TC\_RELAY(n1, m) \end{aligned} \tag{9}$$

For instances the message integrity property allows a node N to ensure the integrity of TC messages exchanged in the MANET. Node N sends, in its TC message, a list of nodes $MPRS\_SET = \{A, B, C, D\}$ that have selected node N as their MPR with the sequence number equal to a value p. The TC message is forwarded by node D and then by node E in order to reach the whole MANET. This case could present two types of possible attacks on the payload of the TC message:

- Modification of the list of MPR selectors: if node D (respectively E) is a lying node and tries to modify the content of the TC message sent by node N, the node N (respectively D) will detect this intrusion.
- Modification on the sequence number: the intruder node D (or E) forwards the received TC message by modifying the sequence number into another value $p'(p' >> p)$. Consequently, nodes E and F stop treating any TC messages originated from A with a value lower than p'.

When a node receives a Hello or TC message, it applies these properties to check the validity of the received message. If the property is violated, then many attacks are possible: (1) The message sender has lied and wished to be selected as a MPR (Link spoofing), (2) The message sender has lied on its identity (identity spoofing) or (3) some relays or an intruder along the way between the source of TC message and the receiving node could also modify the message.

The properties (defined in Basic properties specification for detection of liar nodes) can only detect a "liar node" described in section 6.1. Unfortunately, our detection process based on these properties works well in the case of "information redundancy". So we investigate other properties to detect selfish profiles, and a way to allow a node to send the detection information to its neighborhood. Now we introduce how we can detect these profiles even if there is no information redundancy.

Interval Transmission Property: When a node A selects node B as MPR. the node B must send a TC message with A inside. The emission interval of TC is defined in the TC message (by default this interval is 5 seconds). So, if the node B does not send this TC message before this delay, the node A can detect a "lazy node" or a "selfish node". All common neighbors of A and B can also detect this profile, because they also received the Hello message from A, and they can check if B sends a TC message. An example is shown in figure 2(a).

$$\begin{aligned} &HELLO(na, nb, m) \wedge IN\_MPRS\_SET(nb, m) \\ &\quad \rightarrow \bigcirc^{\leq TC\_INTERVAL} TC(nb, nv, m') \end{aligned} \tag{10}$$

In the same way, a node can detect if a TC message is forwarded or not before a $TC\_INTERVAL$. The TC message is broadcast in the whole network by the MPR node. So when a MPR node forwards a TC message, this node checks if this message is forwarded before some delay. If this message is not forwarded the node detects the "selfish node". However the presence of ambiguous collisions, receiver collisions, limited transmission power, collusion, and partial dropping are detected as "selfish node" whereas they are not. But if there is a collusion, or a limited transmission power, or partial dropping. Therefore the source is not very sure that the packet arrives to the destination. Thus, it is better to change the intermediate nodes to reach the destination.

$$\begin{array}{c} TC(na, nb, m) \wedge MPR\_NEIGHBOR(na, nb) \\ \rightarrow \bigcirc^{\leq TC\_INTERVAL} PROPAGATE(nb, nv, m) \end{array} \tag{11}$$

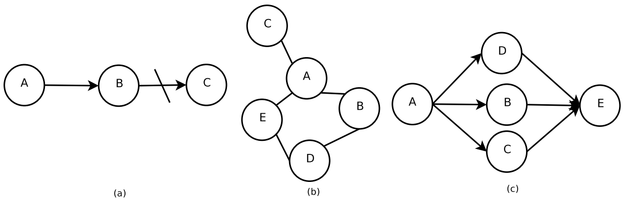

Fig. 2. (a) An example of a "selfish node". (b) No node can check if the node C is really a neighbor of B . (c) To reach the node E, the node A chooses the node which has the greater willingness.

Neighbor Relationship Property: When the local node A has a 2-hop node C reachable by only one neighbor B, it means that B is a MPR of A. But there is no way to know if this link between B and C exists. To avoid the impact of B, the node A decreases the confidence in node B. Thus, the node A chooses other MPR whose confidence is higher. Notice that the node B is still a MPR of A. An example is shown in figure 2(b).

Willingness Property The willingness field is a parameter exchanged in the Hello message [4], thus only the 1-hop neighbors receive this parameter. The willingness field has a value by default, so when a new node arrives in the network it has the default value therefore the new node is not isolated, its (control) messages are exchanged or forwarded. However, when a node finds an intrusion, the node informs about this detection according to the willingness field. So when the willingness is low, it means that the node has incorrect or malicious behavior. When a node A receives an information from the node B, where the node C is claimed as a malicious node (because its willingness is low). the node A applies a reputation function (see below) to deal with this information. Our function is based on [5] that we modify to be in compliance with our topic. The willingness is computed every time the node receives a Hello or TC message.

$$w_x = w_1 + w_2 \tag{12}$$

Where w_1 is the checks over the properties defined in this section and w_2 is the detection information from its neighbors.

$$w_1 = w_{Lx} * p \tag{13}$$

$$w_2 = 1/N * (\sum_{k=0}^{N} (w_{Lk} * w_{kx})/w) * (1-p) \tag{14}$$

With:

- w_x is the final willingness in the neighbor x. This willingness is used to choose the MPR nodes.
- w_{Lx} is the willingness of the local node in the neighbor x. If these properties (defined in 6.2) are violated by the node x, then the willingness changes. We only choose to take into account the most recent information about the nodes. Thus, every time the node receives a message, this value decreases if a property is violated. w_{Lx} increases if no property is violated.
- w_{Lk} is the willingness of the node in the neighbor k. The greater w_{Lk} is, the greater w_{kx} impact is.
- w_{kx} is the willingness of the node k in the neighbor x
- w is the willingness by default defined in OLSR specification.
- N is the number of neighbors
- p and $(1-p)$ is the weighting. Here, $p \geq 0,5$, and $(1-p) = Min\langle(\lfloor N/3 \rfloor); 0,5\rangle$. Where $(1-p)$ is the minimum between the number of neighbors divided by three and $0,5$. If a node has less than three neighbors, we do not take into account the information from the neighborhood, because we do not have enough neighbors to make a good average of the willingness. Thus, more the node has neighbors, more they have influence.

In the RFC of OLSR [4], the 2-hop neighbors is the only parameter to select the MPR node. With our approach the willingness and the 2-hop neighbors help the choice. Hence, the choice is better and allows the local node to keep away the intruders. In Figure 2(c), the node A must choose one of B, C, D to be its MPR. For that, A chooses the node which has the greater willingness. If the node A does not detect any intrusion, the MPR is randomly selected. To avoid this issue, the node takes into account the detection information from its neighbors.

6.3 Our Algorithm For Profile Identification

Our mechanism can be implemented on each OLSR node in order to detect conflicts or inconsistencies in the OLSR control messages. When a node receives a TC message, it uses these properties to check and validate the TC messages before it updates its routing tables. A node which sends TC message and uses these properties to detect if there is an anomaly during the exchange of routing information. By doing so, the security level and the robustness of routing operation can be optimized. To allow normal node to choose another path without intruders.

For this purpose, we weave the properties defined in section 6.2 in the code using an AOP approach. As each property is checked when a message is received, a property is used to identify some pointcuts. We weave at these pointcuts our detection and reaction mechanism in the same manner as we did in wired networks [6] for securing the TCP/IP protocol.

In the weaving approach (see figure **??**), the functional aspect is the OLSR protocol, especially the message reception specification part.The information about the control message is used to update the Topology table, and then the MPR nodes and routes are chosen.

As long as there is no detection, the OLSR algorithm does not change (see figure 3). When a willingness is updated or a property is violated, the algorithm for profile identification is applied. First, our algorithm checks the properties defined in section 6.2. The algorithm then identifies the node profile and computes the willingness according to this profile. The last aspect in figure 3 changes the list of nodes using computation of willingness and then to see to it that only non malicious nodes are chosen. Sending TC or Hello message with the new willingness, the neighbors can take this information into account.

Our mechanism checks several properties to find a reliable path to the destination. However the complexity of this algorithm is in $O(n^2)$ where n is the number of 1-hop symmetric neighbors, but we optimized this complexity using the hashmap to get a complexity in $O(1)$. Therefore, when a node A receives a control message from node B, it checks if the 1-hop symmetric nodes identify B as an 1-hop symmetric node. Therefore the time cost is $n * O(1) = O(n)$. Moreover, the maximum interval transmission between two control messages is the interval transmission between two TC messages, by default this interval is 5 seconds [4]. Therefore a detection is made and communicated in less than 5 seconds.

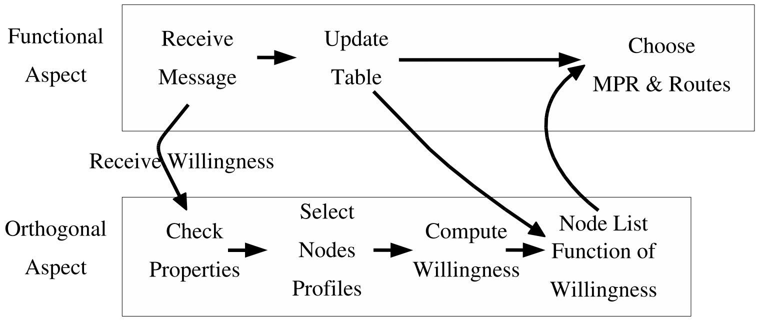

Fig. 3. Weaving functional and orthogonal aspects

7 Availability Experimentation In OLSR

We simulate a Mobile Ad hoc NETwork (MANET) in section 7.1 to test our algorithm. In section 7.2, we illustrate how the algorithm works, and which attacks it can detect. Also, we show that a node can reach another node even in the presence of a malicious node. In section 7.3, we deal with some extreme cases, where a malicious node succeeds in preventing some node from reaching another node.

7.1 Experimental Virtual Network

To simulate a network in our experimentation, we use the User-Mode Linux [9] to create nodes and Iptables [10] to make the links between nodes.

User-Mode Linux provides a virtual machine. User-Mode Linux is a safe, secure way of running Linux versions and Linux processes.

After creating the nodes, we simulate the physical link between them. For this purpose, we use Iptables to write rules that accept packets from the neighbor nodes and block other traffic.

Thanks to User-Mode Linux and Iptables we obtain the network presented in figure 2(c). We choose this topology to test the impact of an intruder like "Liar node", "Selfish node". We assume that the links between are not noisy and the nodes are not very mobile. If there are noisy, this problem is dealt with by the lower layer or the routing protocol itself. If the nodes are mobile therefore the nodes generates incorrect information and it would be selected as a liar node in a first time. After a delay it would not generate other incorrect information and its willingness would be greater and it would be selected as a normal node by its neighbors. The main of our approach is to select the good neighbor to forward the message.

7.2 Analysis

We present some results of our simulations using the example of figure 2(c). The simulation results show the contents of routing tables for each node of the chosen topology: (1) activated analysis mechanism and (2) deactivated analysis mechanism. We then analyze the topology with the normal node behavior, and finally the topology with an intruder and without the analysis mechanism, and the topology with an intruder and the analysis mechanism.

Normal Node Behavior Simulation: We started our simulation with the normal behaviors of nodes without any attack and any verification, and figure 2(c) summarizes the routes used in the network. In this case, we supposed that the quality of all the network links was perfect (without packet loss). The choice between 2 neighbors which have the same 2-hop neighbors is random, because there is no other parameter to help the choice. Table 1 presents relevant OLSR data obtained for the example.

Node	1-hop	2-hop	MPR	MPR
	neighbors	neighbors		selectors
A	B,C,D	E	D	C
B	A,B,C,E	-	-	D
C	A,B,E	D	A	-
D	A,B,E	C	B	A,E
E	B,C,D	A	D	-

Table 1. Relevant OLSR data

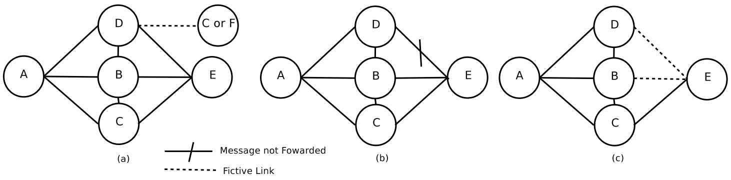

Fig. 4. in (a) the node D creates a link with a known or unknown node, in (b) the node D is a selfish node, in (c) The node D and B make a wormhole attack

Attacks Simulation: In figure 4, in the first case, the nodes A, B and E chose only the node D as MPR because D has created a fictive link with a known or unknown node. In the second attack, the node D does not forward the message from A to E. The node A and common neighbors of A and D do not detect this attack and the node D stays a MPR node. In the third case, the nodes D and B make a wormhole attack, and they claim that the node E is their neighbor. In this case, the node D stays the MPR of A, and the node C chooses the node B as MPR. So the node B and D become MPR. Thank to this attack, the node B and D obtain privilege.

Use Of The Analysis Mechanism: In this step, all nodes (except the intruder) run the same OLSR protocol in which our detection and response mechanism is implemented. In figure 4, in the first case the nodes A, B and E detects that the node D has a link with an unknown node (Neighbor relationship property defined in 6.2). So they decrease the willingness in the node D and thus choose another MPR. So The node A and E chose B or C as MPR. But the node D stays the MPR of A, B and E because the node F is its neighbor. This limits the impact of the attack because the nodes chose another node as MPR to reach their 2-hop neighbors.

In the second attack, the node D does not forward the message from A to E. In this case the node A and B detect that the node D is a "selfish node" (Property of transmission intervals defined in section 6.2). Therefore the node A chooses the node B or C as MPR. Decreasing the willingness of D, the node B sends this detection information to the node E and the node E computes the new willingness (Willingness property defined in section 6.2) and chooses B or C as MPR. We obtain the same result when "liar node" occurs, for example if the node D claims that C is in its neighborhood. This example shows that our approach provides means to choose the good MPR despite the presence of "selfish node" or "liar node".

In the third case, the node C detects that the node B is a "liar node" (Hello-TC relationship or/and MPR-MPR selector relationship property defined in section 6.2), and sends this detection information to the node A. The node A decreases the willingness of the node B (Willingness property defined in section 6.2). So the node A chooses between C and D to be its MPR. At time t_0, the choice is random but at time t_1 the node A chooses the node C. The node C sends a TC message from E to A and the willingness of A in C increases and is greater than B. If the node A has more neighbors, this makes easier the choice of a good MPR, so at time t_0 the choice is only C or another good node. This shows that, using our approach, the wormhole attack does not prevent a non malicious node from reaching another node.

If a route exists between two nodes, then the nodes are reachable even if an intruder tries to change or to block the route. We plan to test in our future works several mobility models using the network simulator ns2 [1].

7.3 Extreme Cases

In this section, we show that there are cases impossible to solve. For instance, if the node has only intruders as neighbors, there is no solution.

Another extreme case looks like the third case in the figure 4, where a wormhole occurs. But here, the willingness of B in C is close to 0. The willingness of B in D and the willingness of B in D are maximal and the node B or D simulates a fictive TC message from E. So the node A chooses D as MPR. However, the problem disappears when there is a larger number of non malicious nodes because the node will take other willingness into account and will choose C or another good node.

8 Conclusion

The techniques presented in this paper are based on specifying security properties in MANET, especially the availability property. If a route exists from a mobile node to another, then this node (if it is permitted) would be able to obtain the route whenever it needs. And the routing operation would take a bounded delay to complete.

Through this study, we chose the OLSR protocol to analyze the availability requirements for MANETs. Several properties related to availability have been expressed based on the specification of the protocol OLSR (these properties are compliant with the RFC3626) and malicious node profiles are used to deploy an intrusion detection and reaction technique. Each MANET node observes its neighbors' behaviors corresponding to the received messages which provides means for checking if its neighbor is malicious or not. If a detection occurs, the node sends this information to its neighborhood. This approach seems to us the most adapted for MANETs. Aspect-Oriented Programming (AOP) makes easier the implemention of availability properties. AOP allows us to keep the standard OLSR specification unchanged when there is no detection and to use our algorithm when a detection occurs. The AOP approach allows us to define a method to secure any routing protocols provided we have specified security aspects to be woven in the protocol. To validate our analysis, an experimentation has been done on a virtual network where the analysis mechanisms and several misbehavior have been implemented. The obtained results from our experiments encouraged us to go further in our investigations. We plan to test our approach in a network simulator to take into account several mobility models. The objective is to express other properties that we shall use to adapt our detection/reaction mechanism.

Acknowledgement: This work was supported by the RED Celtic project and a French DGA (Direction Générale de l'armement) grant.

References

1. A collaboratoin between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC. *The ns Manual.* http://www.isi.edu/nsnam/ns/doc/index.html.
2. C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, and D. Raffo. Securing the OLSR protocol. In *2nd IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2003)*, Mahdia, Tunisia, June 25–27 2003.
3. S. Bhargava and D. P. Agrawal. Security enhancements in aodv protocol for wireless ad hoc networks. In *VTC 2001 Fall*, page vol. 4 pp. 214347, 2001.
4. T. Clausen (ed) and P. Jacquet (ed). Optimized Link State Routing protocol (OLSR), October 2003. RFC 3626, http://www.olsr.org/.
5. M. Conrad, T. French, W. Huang, and C. Maple. A lightweight model of trust propagation in a multi-client network environment: To what extent does experience matter?. In *ARES*, Vienna, Austria, 2006. IEEE Computer Society.
6. F. Cuppens, N. Cuppens-Boulahia, and T. Ramard. Availability Enforcement by Obligations and Aspects Identification. In *ARES*, Vienna, Austria, 2006.
7. F. Cuppens, N. Cuppens-Boulahia, and T. Sans. Nomad : A Security Model with Non Atomic Actions and Deadlines. In *The computer security foundations workshop (CSFW)*, Aix en Provence, France, 2005.
8. E.W Dijkstra. A note of two problems in connection with graphs. In *Sumerisclie Mathematik*, pages vol.1. pp. 269–271. 1959, 1959.
9. D.P. Bovet and M. Cesati. *Understanding the Linux Kernel.* O'Reilly, 2003.
10. Gregor N. Purdy. *Linux Iptables Pocket Reference.* O'Reilly, 2004.
11. F. Hong, L. Hong, and C. Fu. Secure OLSR. In *19th IEEE International Conference on Advanced Information Networking and Applications (AINA '05)*, Tamkang University, Taiwan, March 28–30 2005.
12. S. Isida, E. Ando, and Y. Fukuzawa. Secure routing functions for OLSR protocol. In *2005 OLSR Interop and Workshop*, Palaiseau, France, July 28–29 2005.
13. G. Kiczales. Aspect-oriented programming. *ACM Comput. Surv.*, 28(4es), 1996.
14. B. Lampson, M. Abadi, M. Burrows, and E. Wobber. Authentication in Distributed Systems: Theory and Practice. *ACM Transactions on Computer Systems*, 10(4):265–310, November 1992.
15. S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. *Mobile Computing and Networking*, pages 255–265, 2000.