Analysis of the implicit trust within the OLSR protocol

Asmaa Adnane[1], Rafael Timóteo de Sousa Jr[2], Christophe Bidan[1], and Ludovic Mé[1]

[1] Supélec - Equipe SSIR (EA 4039) {aadnane, cbidan, lme}@rennes.supelec.fr
[2] University of Brasília - LabRedes desousa@unb.br

Abstract. Trust is an interesting criterion for analyzing and comparing network protocols. The goal of this paper is to explicit the different types of trust relations between entities which exchange routing information and establish a routing infra-structure based on the OLSR protocol. One such entity assumes the other entities will behave in a particular way and the relations coming from this trust behavior are expressed in this paper using a formal language. This approach highlights the process of trust construction in OLSR and allows the analysis of trust requirements for this protocol, as well as the expression of attacks related to the betrayal of trust relations. Besides, this analysis allows the description of indicators for OLSR entities to have a protective mistrust behavior when effectively acting based on trust relations.

1 Introduction

Several research studies were conducted the last few years aiming at developing protocols for networks whose nodes communicate directly with each other to relay messages without the support of a central entity. This operating mode characterizes the ad hoc networks, for which the Internet Engineering Task Force (IETF) standardized some routing protocols such as the Optimized Link State Routing Protocol (OLSR) [3].

The objective of this paper is to identify and formalize trust assumptions that are implicitly used by the OLSR protocol. One of the goals of this analysis is to propose extensions to OLSR in order to make it more flexible to the variations of the environment and more resistant against security treats, while avoiding excessive restrictions on the auto-organization capacities and the dynamics of the network.

For this purpose, we begin from the idea of trust classification, which consists of a delimitation of the circumstances where a trust relationship is established, and we analyze the classes of trust present in OLSR. Initially, we present the language used to formally express trust clauses and the definition of trust subjacent to this language. Then, we expose the general characteristics of the OLSR protocol and its security problems. Finally, we present the OLSR implicit trust clauses and analyze the attacks against this protocol according to these implicit clauses.

Please use the following format when citing this chapter:

Adnane, A., de Sousa, R. T., Jr., Bidan, C. and Mé, L., 2007, in IFIP International Federation for Information Processing, Volume 238, Trust Management, eds. Etalle, S., Marsh, S., (Boston: Springer), pp. 75–90.

The paper is organized as follows. Section 2 surveys related research works. Section 3 presents the formal language for the expression of direct and derived trust clauses. The OLSR protocol is briefly described in section 4. The analysis of implicit trust within OLSR is presenteted in section 5. Section 6 is a study of some attacks against OLSR, from the point of view of trust. Finally, the conclusion sumarizes the results and indicates possible directions for future research.

2 Related works

The concepts of trust, trust models and trust management have been the object of several recent research projects. Trust is recognized as an important aspect for decision-making in distributed and auto-organized applications [5] [4]. In spite of that, there is no consensus in the literature on the definition of trust and what trust management encompasses. Many authors propose their own definitions of trust, each one concerning a specific research domain [13]. As a result, a multitude of formal models for trust calculation and management emerged, but this also lead to similar concepts appearing under different names and reciprocally [13] [14]. To mitigate this situation, in this paper we use the trust definition and a language to express trust proposed by [4], which permit to formalize and clarify trust aspects present in communication protocols.

A framework for specification and security analysis of mobile wireless networks communication protocols was proposed by [16], specifically for studying the SAODV routing protocol. However this study was not interested in the concept of trust. Other studies treat trust management and its relation to the routing operation in ad hoc networks [10],[7],[3].

Other authors [10] explored this subject to solve the problem of cooperation (one of the concepts related to trust) in ad hoc networks and to constrain the selfish nodes to cooperate.

The CONFIDANT project [7] proposes an extension of the DSR protocol providing nodes with a special component to observe and adapt to the behavior of the neighbors according to their reputation. Paper [9] proposes an extension to DSR, which selects the route based on a local evaluation of the trustworthiness of all known intermediary nodes on the route to the destination. The TRANS protocol [11] proposes a location-centric architecture for isolating misbehavior and establishing trust routing in sensor networks.

Reference [17] proposes a mechanism of anonymous signature bound to the record of a node's interactions history and associated to a proof of proximity, in order to allow the establishment of trust relations based on the history of interactions among the concerned nodes.

The OLSR specification [3] does not establish any special security measures, but recognizes that, as a proactive protocol, OLSR is a target for attacks against the periodic broadcast of topological information. Several efforts were made with the objective of finding security solutions for OLSR [6] [7] [8]. A survey on these solutions is presented by [6] which proposes a security architecture based on adding a digital signature to OLSR control messages, together with methods to validate the actual link state of nodes and to control intra-network misbehavior. One of these methods is based on the use of a protocol to detect misbehaving nodes using a reputation evaluation system. Other more traditional solutions, based on cryptography and authentication, are developed in [2].

However, these proposals do not provide an analysis of implicit trust in ad hoc routing protocols (especially OLSR), which is the basic contribution of this paper.

3 Expressing trust relationships

We use the language proposed by [4] for expressing the clauses concerning trust in a networking protocol. The concept of trust subjacent to this language is expressed by the fact that if an entity A trusts an entity B in some respect, informally means that A believes that B will behave in a certain way and will perform some action in certain specific circumstances.

The trust relation is taken into account if the possibility of realization of a protocol operation (the action) is evaluated by entity A on the basis of what it knows about entity B and the circumstances of this operation. According to the considered action and its circumstances of execution, it is necessary to distinguish various trust classes as defined by [4] and [13], so, for the sake of precision on the formalization of trust relations required by OLSR, in section 5 we propose appropriate classes to the actions performed by this protocol, such as the trust in another entity to route messages (routing trust). Still in accordance with [4], we distinguish the direct trust relations and the derived trust relations, the last ones being established from recommendations of other entities. Given the presence of several types of entities in the execution environment of a protocol and the existence of indirect relationship between the entities, it is necessary to distinguish these two types of trust relations. Thus, the clauses relating to trust are expressed with the following notations:

- each entity is identified by a single name; the terms A, B, C indicate specific entities, while the terms R, S indicate sets of entities;
- a specific class of trust is noted cc;
- the expression $A\ trusts_{cc}(B)$ means that A trusts B with respect to the action cc;
- $A\ trusts_{cc}(S)$ means that A trusts the set of entities S with respect to action cc, S being defined as the set of all entities for which a certain predicate holds;
- $A\ trusts_{cc-C}(B)$ means that A trusts B to perform action cc with respect to the entity C (but not necessarily to other entities);
- $A\ trusts.rec_{cc}(B) when.path[S] when.target[R]$ means that A trusts the recommendations of entity B about the capacity of other entities to perform

action *cc*. The *when* clauses allow the specification of constraints on the recommendations. The trust recommendation *path* is a sequence of entities such that each one is recommended by its predecessor, so the *when.path* specifies the only set of entities to be considered, at each point in some trust recommendation path, as candidates for the next step in the path. The *target* clauses specifies the only set of entities to be considered as candidates for becoming target entities in some recommendation path.

In the following sections, the use of this language, together with the mathematical set theory, allows us to reason about the trust required by the OLSR protocol and to explicitly express trust relations between the entities executing this protocol. This formal approach also has the interest to allow the analysis of certain attacks against OLSR by revealing the implicit trust relations these attacks exploit.

4 Characteristics of the OLSR protocol

OLSR is a proactive link-state routing protocol, which uses an optimized flooding mechanism to diffuse partial link state information to all network nodes. The protocol uses multi-point relays (MPRs) which are selected nodes that forward broadcast messages during the flooding process. The link state information is generated only by nodes elected as MPRs and each MPR must only report on the state of links between itself and its selectors. Two types of control messages, HELLO and TC, allow each node to obtain and declare network topological information.

HELLO messages are sent periodically by a node to advertise its links (declared as asymmetric, symmetric or MPR) with neighbor nodes. Received HELLO messages allow a node to memorize information about links and nodes within its 2-hop neighborhood, so as to constitute the internal mental state of each node, which is represented in the form of sets, including the link set (LS), the neighbor set (NS), the 2-hop neighbor set (2HNS), the set of nodes selected as MPR (MPR Set - MPRS) and the set of neighbor nodes who chose the node as MPR (MPR Selector Set - MPRSS). These sets are updated and used continuously for MPR selection, in such way that a message sent by the node and relayed by its MPR set (i.e., elements of its MPRS) will be received by all its 2-hop neighbors. Each node also records the addresses of its neighbors who selected it as MPR (what constitutes the MPRSS). Thus, HELLO messages allow a node to establish its view of the "small world" (within the 2-hop neighborhood).

The TC message conveys the topological information necessary for computing routes to the whole network, the "big world". The reception of TC messages allow a node to obtain information about destination nodes and to keep this information in its Topology Set. A node which was selected as MPR periodically broadcasts TC messages advertising symmetric neighbors and these messages are flooded in the whole network allowing the nodes to compute the topology to be used for routing (routing table).

With regard to the security aspects, the RFC 3626 does not specify any security measures for OLSR, even though this RFC describes the vulnerabilities of the protocol. The principal security problems are related to the facts that the topology of the network is revealed to anyone who listens to OLSR control messages, that nodes may generate invalid control traffic, that interferences may come from outside the OLSR environment and that the protocol operations assume the unicity of the IP address to identify a node. Traditional solutions based on cryptography and digital signature of messages, authentification of the origin and time-stamping of the messages, as well as address restriction and filtering, are indicated in the standard to mitigate these security problems. An implementation of these solutions is presented by [2]. Still, it should be noted that trust is not treated by this reference.

Given the general description of the protocol and the definition of the sets maintained by the OLSR node, it is possible to use the language described in section 3 to express the trust relationships in this protocol. Generally, the nodes (N) are considered to be cooperative and to trust the fact of obtaining the cooperation of the neighbor nodes. This behavior corresponds to the concept of general trust as defined by [5]. For example, the RFC 3626 [3] states that "a node should always use the same address as its main address" (p. 5), which is the basic belief of a node in the identity of others. This statement is translated using the formal language presented in section 3 to the expression:

$$N_i \ trusts_{id}(N_j), i \neq j$$

In the same way, other similar expressions are employed in the following sections for the analysis of the implicit trust required by OLSR and the description of attacks against OLSR.

5 Analysis of OLSR implicit trust aspects

In this section, while expressing the implicit trust rules in OLSR, we present reasonings on trust which could be used for selecting the MPRs of a node and for computing its routing table. We show that trust, if "explicitly" expressed, can be a reasoning factor of a node about its small world and about the routing towards the big world.

The node collects information about link configuration (small world) and routing topologies (big world) from the exchanges of HELLO and TC messages, respectively. The analysis below allows us to extract OLSR implicit trust rules and to suggest that the protocol should also integrate the concept of mistrust towards its choices of MPR and routes. For this purpose, we use the following definitions:

- *MANET*: the set of the whole MANET nodes,

- LS_x (Link Set): the link set of the node x,
- NS_x (Neighbor Set): the set of symmetric neighbors of the node x,
- $2HNS_x$ (2-Hop Neighbor Set): the set of 2-hop neighbors of the node x,
- $MPRS_x$: the set of nodes selected as MPR by the node x ($MPR_x \subseteq NS_x$),
- $MPRSS_x$ (MPR Selection Set): the set of symmetric neighbors which have selected the node x as MPR,
- TS_x (Topology Set): the set containing the network topology as seen by the node x,
- RT_x (Routing Table): the routing table of the node x.
- $dist : MANET^2 \rightarrow \aleph$: the function which provides the distance, expressed as the number of hops, between two nodes of the network.

The following sections present the evolution of a node's trust during the operations of link sensing, MPR selection (computation of MPRS), MPR signaling (computation of MPRSS), and routing table calculation.

As indicated before, initially the nodes are generally trustful [5], since they do not know anything on their environment and believe in all information that they receive from others without checking its validity.

5.1 Discovering the neighborhood - Link sensing

Initially a node X does not know any neighbor, therefore it does not have any view of the network. The node starts to build its view with the reception of HELLO messages coming from the neighbors. We note $X \overset{HELLO}{\leftarrow} Y$ as the reception of a HELLO message coming from Y. Firstly, these messages allow the node to detect asymmetrical links, leading to a modification of the mental state of X about its trust in node Y, i.e., X knows Y but does not trust it yet, because X is not sure that Y functions in accordance with the OLSR specification, with regard to the reception and sending of HELLO messages:

$$X \overset{HELLO}{\leftarrow} Y,\ X \notin LS_Y \Longrightarrow X \neg trusts\ (Y) \qquad (1)$$

This expression means that X does not trust Y neither to be a symmetrical neighbor, nor to be a MPR, although X receives HELLO messages from Y. However, being an agent generally trustful [5], X diffuses HELLO messages that can be received by Y, which in turn will be able to take them into account and to add X to its set of symmetrical neighbors NS_Y.

If Y acts according to the protocol, i.e., if it sends HELLO messages informing that it has a link with X, then a new situation of trust is reached:

$$\begin{aligned} X \overset{HELLO}{\leftarrow} Y,\ X \in LS_Y \Rightarrow X\ trusts_{ID \cup NI}(Y), LS_X &= LS_X \cup Y \\ 2HNS_X &= 2HNS_X \cup (NS_Y - X) \end{aligned} \qquad (2)$$

A trust relation has just been built which is concretized by the fact that now X regards Y as its symmetrical neighbor, and the symmetrical neighbors of Y as 2-hop neighbors. In addition, this trust relation is seen as symmetrical, since Y is expected to behave in the same way as X:

$$Y \overset{HELLO}{\leftarrow} X \Rightarrow Y\ trusts_{ID \cup NI}(X)$$

This symmetrical relation is the base for future decisions which will be taken by X about its small world (MPR selection), but also, indirectly, for the routing towards the big world (calculation of the routing table) through the exchange of TC messages.

5.2 MPR selection - Computing the *MPRS*

In OLSR, the only criterion for MPR selection by a node X is the number of symmetrical neighbors of a candidate node Y, which defines the degree of Y, noted $D(Y)$ and calculated by the formula:

$$\forall Y \in NS_X :\ V_Y = NS_Y - NS_X - \{X, Y\}, D(Y) = card\{V_Y\} \tag{3}$$

Firstly, the choice concerns the MPRs for relaying to nodes in the 2-hop neighborhood that can be reached only through paths including the chosen MPRs:

$$\begin{aligned} MPRS_X = MPRS_X \cup \{Y \in NS_X : \exists Z \in 2HNS_X : Z \in NS_Y, \\ \forall V \in NS_X : Z \notin NS_V\} \end{aligned} \tag{4}$$

Then, while there are nodes in $2HNS$ which are not covered by at least one node in the MPR set, this set is extended with other MPRs whose selection is based on their reachability to the maximum number of nodes in $2HNS$ (in case of multiple nodes providing the same reachability, the node whose D(Y) is greater is selected as MPR) until all nodes in $2HNS$ are covered:

$$\begin{aligned} \exists V \in 2HNS_X : \forall Y \in MPRS_X :\ V \notin NS_Y \Longrightarrow MPRS_X = \\ MPRS_X \cup \{Y \in NS_X : D(Y) = MAX\{D(Z) \forall Z \in NS_X\}\} \end{aligned} \tag{5}$$

In terms of trust, this means that X trusts the nodes in its MPR set for routing:

$$\forall\ Y \in MPRS_X : X\ trusts_{fw}(Y) \tag{6}$$

Consequently, the nodes in $MPRS_X$ are required to recommend to X the routes to the distant nodes:

$$\forall\ Z \in MANET : X\ trusts.rec_{fw}\ (Y)\ when.path[MPRS_Y]\ when.target[Z]$$

Considering that the nodes $MPRS_Y$ themselves trust other MPRs, the route from X to Z is formed by a sequence in the form of the predicate: $route_{Y_1 \to Y_n} = Y_1, ..., Y_n$ with $Y_{i+1} \in MPRS_{Y_i}$, which allows to extend the expression above to obtain:

$$\forall Z \in MANET : X\ trusts.rec_{fw}(Y)\ when.path[route_{Y \to Z}] when.target[Z] \tag{7}$$

This expression presents the general rule of trust recursivity for the routing in the networks operating under OLSR.

5.3 MPR Signaling - Computing the $MPRSS$

This calculation allows a node X to discover information about the trust that other nodes place on X itself. The calculation of the $MPRSS_X$ is expressed by the following formula:

$$X \overset{HELLO}{\leftarrow} Y, X \in MPRS_Y \Rightarrow MPRSS_X = MPRSS_X \cup \{Y\} \tag{8}$$

As X allows the nodes of its $MPRSS$ to use its resources for routing, which constitutes a form of access trust as discussed in Section 3, the calculation of $MPRSS_X$ implies that X trusts Y to use X resources for routing without causing any harm and also that X trusts Y for advertising that X is a MPR. These trust relations correspond respectivelly to the following expressions:

$$X\ trusts_{at}\ (Y), X\ trusts_{dt}\ (Y)$$

5.4 Computing the routing table

The routing table is computed from the information contained in the local link information base and the topology set. Therefore, the routing table (RT) is recalculated if the node detects a change in either of the sets LS, NS, $2HNS$, TS, $MPRS$ or $MPRSS$.

Each entry in RT consists of: (*R_dest_addr*, *R_next_addr*, *R_dist*, *R_iFace_addr*), and specifies that the node identified by *R_dest_addr* is located *R_dist* hops away from the local node, that the symmetric neighbor node with interface address *R_next_addr* is the next hop node in the route to *R_dest_addr*, and that this symmetric neighbor node is reachable through the local interface with the address *R_iface_addr*.

Each node X has its view of the network topology and selects the shortest path to reach any other node Z passing through a selected MPR Y. The routing table is thus computed using a shortest path algorithm [18]. From the point of view of trust,this calculation will allow X to trust Y for the routing towards Z. If we note $T = (Z, Y, N, I)$ for a tuple of RT_X, the following relation is obtained:

$$\forall T \in RT_X \Rightarrow X\ trusts_{fw-Z}(Y)\ or\ X\ trusts_{fw-R\_dest\_addr}\ (R\_next\_addr) \tag{9}$$

Moreover, the routing table is calculated so that there is only one route towards each destination:

$$\forall X, Z \in MANET,\ Z \notin NS_X \Rightarrow \exists!\ T \in TR_X :\ T.R\_Addr\_Dest = Z \tag{10}$$

and each selected route is the shortest among the routes starting from MPR nodes, which defines a predicate that we call $MinDist(X, Z)$:

$$Y \in MPRS_X :\ MinDist(Y, Z) = MIN\{dist(A, Z)/A \in MPRS_X\} \\ \Rightarrow T.R\_Next\_Addr = Y \tag{11}$$

The inherent risk in the choice of only one route towards any destination is to choose, as router, a corrupted or misbehaving node. In the following section, we explain how this vulnerability can be exploited by the attackers, who give false information about the network topology in order to direct all the traffic of the network towards them and/or to disturb the operation of the protocol.

According to the expression (11), even if there are several paths towards Z, X will only choose the shortest route starting from one of its MPR. The routing table calculation is a reasoning based on the distance and results in the set of routes which the node considers as the most adequate for the routing. Actually, the goal of this calculation is to suitably choose the MPRs among those which offer routes towards the destinations. After computing the distances to destinations, the node will place more trust in those nodes which offer the shortest paths towards the destinations (9).

The selection of Y as MPR by X for routing towards a node Z implies that X, not only trusts Y for routing (6), but also trusts the choices of the routes made by Y (7). Actually, there is a chain of this indirect trust relation between X and any relay forwarding the packets to Z and this chain has the particularity that only the last relay before Z, being a MPR of this target node, exchanges control messages directly with Z (HELLO messages). This sequence expresses the transivity of MPR recommendations in OLSR, a property which allows us to use the deduction algorithm presented by [4] to obtain the following trust relation:

$$X \; trusts.rec^{*}_{fw-Z} \, (Z) \; when.target[Z] \; when.path[Z] \tag{12}$$

This expression means that the routing target node is itself the starting point of the trust chain, and its *MPRS* should be properly chosen so that every other node can correctly communicate with this node.

That suggests the existence of a spreading of the trust placed in the MPR. Certain attacks against OLSR exploit the vulnerability resulting from the absence of validation of this derived trust chain. The node should have a degree of mistrust concerning the information used for the calculation of the routing table. This mistrust could be associated to the use of a procedure for validating the routing information which is spread in network (TC messages).

Two results are put forward by this analysis. In the first place, the operations of OLSR generate information related to trust and present implicit trust rules that, as such, are not taken into account by the nodes, but which can be actually exploited to contribute to the security of the protocol. Secondly, the analysis shows that the nodes create trust relationships without validated evidence, not measuring the consequences of these relationships and thus without any mistrust in their choices.

6 Trust-based synthesis of OLSR vulnerabilities

With respect to routing choices, the OLSR reasoning is aimed at calculating the routing table, a behavior that implies thereafter the implicit use of trust relationships between nodes. In other words, OLSR effectively generates information about trust between nodes, but the nodes firstly cooperate and then, without any validation, implicitly deduce information about the other nodes in which they have to trust. The only criterion for this reasoning is the distance between the nodes, an aspect of which they should be careful. Otherwise, mistrust would be a more appropriate behavior in the beginning of a relationship which can lead to cooperation with mischievous nodes. Moreover, the information related to trust is obtained, but is neither used for the future cooperations, nor exploited to improve the operation of the protocol.

To accept information that comes within the received messages, without using a security mechanism (i.e., authentication) or a validation procedure (i.e., checking the protocol logic), is the principal vulnerability exploited by certain attacks against OLSR. These attacks are analyzed hereafter, considering the trust clauses that were explicitly expressed for OLSR in section 5.

In a previous work [15], we proposed a classification (table 1) of these attacks against OLSR. Any node can either modify the protocol messages before forwarding them, or create false messages or spoof an identity, and each one of these actions can be at the base of an attack. As the HELLO message is sent to the 1-hop neighbors and is not relayed, this message is not prone to modification attacks, but rather to fabrication attacks. On the other hand, the TC message is sent to all the network and can thus be used either for modification and fabrication attacks (before the relaying).

Table 1. Vulnerabilities of the OLSR Protocol

Attack	OLSR message	Falsified Routing Information	Origin information in the Corrupted Message
Fabrication	HELLO	Neighbor List	Any
Fabrication and impersonation	HELLO	Link-status	IP Address of the impersonated node
Fabrication	TC	MS list	Any
Modification and impersonation	TC	Sequence Number	Originator IP Address

6.1 Attack 1: Fabrication of HELLO Messages

In this attack (figure 1), the adversary wants to be selected as MPR and fabricates a HELLO message advertising all the nodes previously announced in any HELLO message it has already received, together with an additional unused address, this one with symmetric link status. On receiving this message, all of the attacker's neighbors choose it as sole MPR (according to the rule 4). Thus all traffic originated in these nodes towards destinations outside the 1-hop neighborhood is then forwarded to the attacker. Before the attack, A chooses B as

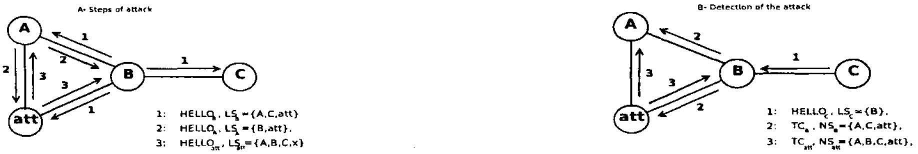
A- Steps of attack
A
B
C
att
1
2
3
1: HELLO_B, LS_B={A,C,att}
2: HELLO_A, LS_A={B,att},
3: HELLO_att, LS_att={A,B,C,x}
B- Detection of the attack
1: HELLO_C, LS_C={B},
2: TC_B, NS_B={A,C,att},
3: TC_att, NS_att={A,B,C,att},

Fig. 1. Hello message fabrication

MPR to transmit data to C. The attack takes place according to the following steps:

1. $att \overset{HELLO}{\longleftarrow} B$: the attacker identifies A and C as neighbors of B;
2. $att \overset{HELLO}{\longleftarrow} A$: the attacker identifies B as a neighbor of A;
3. After receiving a new HELLO message from B, the attacker fabricates a HELLO message announcing $LS_{att} = LS_A \cup LS_B \cup X = \{A, B, C, X\}$ (X is an additional fictitious address announced with symmetric link status).

In consequence of this attack, according to the rule (4), A and B will select att as MPR:

$$A\ trusts_{fw}(att),\ \ B\ trusts_{fw}(att)$$

The attacker acquires the trust of A and B which will choose it for routing towards any node Z in the network, without having a proof of the existence of a path between the attacker and Z. In this example, A will select the attacker to route towards C because it seems to A that this is the shortest path (11), leading to the situation expressed by the rule (7):

$$A\ trusts.rec_{fw}\ (attacker)\ when.path[route_{attacker \rightarrow C}]\ when.target[C]$$

The fact that there is no path $route_{attacker \rightarrow C}$ proves that the nodes A and B should mistrust the information provided in the HELLO message. A trust-based reasoning allows the nodes to check the validity of the topological information so that the nodes A and B can detect the attack without calling upon heavy cryptographic mechanisms.

One of the possible verifications consists in reasoning based on subsequent TC messages. Before the attack, A held B as MPR for routing messages to C and C held B as MPR for routing messages to A, thus $MPRS_A \cap MPRS_C = \{B\}$. After the attack, since B remains as a MPR of C, it will broadcast a TC message advertising C as a symmetric neighbor. In the other hand, the attacker will also broadcast a TC message advertising A and B as neighbors. The reasoning from the point of view of A will lead to contradictory conclusions. By receiving a TC message from B, A will deduce:

$$A \overset{TC}{\leftarrow} B,\ NS_B = \{A, att, C\} \Rightarrow \exists\ Z \in NS_B :\ B \in MPRS_Z$$

To the contrary, node A, after receiving a TC message from the attacker, will also deduce:

$$A \overset{TC}{\leftarrow} att,\ NS_{att} = \{A, B, C, X\} \Rightarrow \exists\ Z \in NS_{att} :\ att \in MPRS_Z$$

To discover the node which selected B as MPR, A reasons by elimination on the set NS_B. Given that $B \notin MPRS_A$ then it is not A which selected B as MPR. To check if it was the attacker that chose B as MPR, A compares the respective neighbourhoods of B and *att*, by checking whether the smallest neighbourhood is included in the largest :

$$[NB_B - \{att\}] \subset [NB_{att} - \{B\}]$$

Then, A deduces that it was not the attacker which selected B as MPR. Thus, it is the node C which did it, establishing the following trust relation:

$$B \in MPRS_C \Rightarrow\ C\ trusts_{fw-A}(B) \tag{13}$$

Moreover, the degree of reachability of B is lower than the degree of the attacker. Thus, based on clause (5), A deduces that C should also choose the attacker as MPR:

$$D(B) < D(att) \Rightarrow\ att \in MPRS_C$$

In terms of trust, C should use *att* for routing towards A:

$$att \in MPRS_C \Rightarrow\ C\ trusts_{fw-A}(att) \tag{14}$$

Considering that it should exist only one route for each destination (10), there is a contradiction between (13) and (14), which leads node A to mistrust the received information, since its view of the network topology indicates that C should have chosen *att* as MPR and not B. The problem arises from the link between *att* and C. Thus, A must be mistrustful regarding these two nodes. But, C should not represent a danger given that it selects B as MPR and thus behaves correctly; on the other hand, the attacker, which was selected as MPR, presents a real risk. The trust-based analysis of this attack shows the importance of message correlation to establish a mistrust-based control in OLSR, according to the following assertions:

- The node has to mistrust another node who declares to be neighbor of all other nodes until these other nodes confirm it effectively as a symmetric neighbor;
- The node has to look for contradictory topological information by correlating the received message contents (HELLO and TC messages).

6.2 Attack 2: Fabrication and Impersonation in HELLO Messages

In this type of attack, the attacker aims at destroying a symmetrical link that exists between two neighbors. After reception of a legitimate message, the attacker generates a spoof HELLO message advertising the link which it wants to destroy with “lost” status. When the target neighbor receives the false HELLO, it will update its link set. Thus no traffic will be forwarded to the target node through the lost link. This attack, which is illustrated in the figure 2, proceeds according to the following steps:

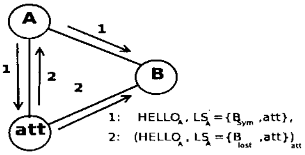
A
B
att
1
1
2
2
1: HELLO_A, LS_A = {B_sym ,att},
2: (HELLO_A, LS_A = {B_lost ,att})_att

Fig. 2. Fabrication and impersonation in HELLO Messages

1. A and B establish a symmetric link by exchanging HELLO messages. Thus, they trust each other (rule 2):
$$B \overset{HELLO}{\longrightarrow} A \Rightarrow A\ trusts_{ID \cup NI}(B),\ A \overset{HELLO}{\longrightarrow} B \Rightarrow B\ trusts_{ID \cup NI}(A) \quad (15)$$
2. by capturing a HELLO from A, the attacker identifies B as a symmetric neighbor of A.
3. after receiving the HELLO message from A, the attacker fabricates a HELLO message impersonating A, advertising B with lost link status. This message makes B alter to asymmetric its link status towards A, thereby blocking any traffic to be forwarded via this link. This implies (according to (1)):
$$B \overset{HELLO_A}{\longleftarrow} att,\ B \notin LS_A \Rightarrow B \neg trusts(A) \quad (16)$$

As OLSR specifies an interval value for the periodic emission of HELLO messages, but does not specify measures to check if messages are received in a very small interval, if this attack occurs, B will continue to receive HELLO messages from A advertising the link to B as symmetrical and spoofed HELLO messages from the attacker declaring the opposite. Thus, B receives two contradictory pieces of information (15 and 16) in a small time interval (lower than the standard interval defined by OLSR), and so must mistrust this information before destroying its trust relationship with A.

The analysis of this attack confirms the potential of the correlation between received messages to establish a control based on mistrust. In the present attack, the node must take into account the factor of time before destroying a trust relationship, according to the following assertions:

- following the reception of a HELLO message advertising a lost link status, the node should not destroy the trust relation and declare the link as lost immediately. It must exchange other HELLO messages to check with the neighbor whether they continue to hear each other;
- as before, the node must mistrust the neighbor who will benefit from the destruction of the link, for example which will be selected as MPR.

6.3 Attacks 3 and 4: Fabrication and modification of TC messages

The objective of these attacks is to provide false network topological information. The attacker fabricates a TC message advertising remote nodes (2 hops or more) as being within its neighbor set (NS). This means that the attacker will be chosen by its neighbors to route traffic to the falsely advertised nodes. A similar outcome can be obtained if the attacker modifies a received TC message. The attacker proceeds according to following steps (Figure (3)):

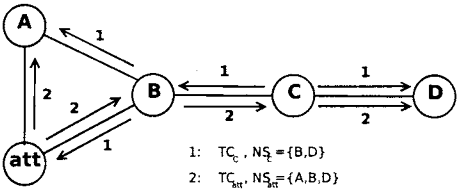

Fig. 3. Fabrication of TC message

1. by receiving a TC Message from C, the attacker identifies D at a distance of 3 hops;
2. the attacker fabricates another TC message, advertising D as part of its NS (symmetric neighbors). This causes A to update its routing table so as to stop routing traffic to D via B and start routing it via the attacker.

In this situation, the attacker takes advantage of the rule (9), leading to the following trust relationships:

$$A\ trusts_{fw-D}(att) \Rightarrow A\ trusts.rec_{fw}(att)\ when.path[route_{att \to D}]\ when.target[D]$$

This is a situation similar to attack 1: the trust relationship above is established without evidence because there is no $route_{att \to D}$. Node A should check this information before changing its routing table. To proceed this verification, A has to wait messages coming from D, which will allow the correlation of network topology information. Initially:

$$A \overset{TC_C}{\longleftarrow} B,\ NS_C = \{B, D\} \Rightarrow \exists Z \in NS_C :\ C \in MPRS_Z$$

Then, A will receive a TC message from the attacker:

$$A \overset{TC_{att}}{\longleftarrow} att,\ NS_{att} = \{A, B, D\} \Rightarrow \exists Z \in NS_{att} :\ att \in MPRS_Z$$

The node A can deduce that: $Z \neq A$, because $att \notin MPRS_A$ and $Z \neq B$, otherwise A would have received the TC_C messages from B and from att. Therefore, $Z = D$, which implies:

$$att \in MPRS_D \Rightarrow\ D\ trusts_{fw-A}(att) \qquad (17)$$

On the other hand, D continues to have C as MPR for routing towards A. Therefore, A will receive data from D via B and will be able to deduce:

$$A \overset{data_D}{\longleftarrow} B,\ D \notin NS_B,\ D \in NS_C,\ C \in NS_B \Rightarrow C \in MPRS_D \qquad (18)$$

According to the rule (9), A can deduce that:

$$C \in MPRS_D \Rightarrow\ D\ trusts_{fw-A}(C) \qquad (19)$$

Given that a node should have only one route towards each destination (10), this expression represents a contradiction with expression (17).

7 Conclusions and future works

The trust aspects of OLSR ad hoc routing could be formalized with the chosen language, which allowed us to interpret attacks against OLSR in terms of trust classes and relations. As a result, we put forward the conditions to use trust-based reasoning as a solution to mitigate certain vulnerabilities of the protocol.

Indeed, the analysis highlights possible measures to render OLSR more reliable and this by means of operations and information already existing in the protocol, without resorting to cryptographic mechanisms. We arrive at the conclusion that a mistrust-based control can be set up to detect suspect behavior using the correlation between information provided in the subsequent received messages. For example, the discovery of neighborhood (link sensing), which is limited to the information provided by HELLO messages, can be strengthened by exploiting the topological information (TC messages) to validate the acquired knowledge and deduce other criteria which a node can use to select its MPR set. Some relationships between nodes can be derived exclusively from a trust-based reasoning. These derived relationships could be used for MPR selection. It is also possible to consider the use of trust as an additional criterion to calculate the routing table, besides the degree of the nodes (number of declared neighbors).

Finally, it is possible for a node to discover the information about the trust the other nodes place on it. By principle, any node could consider the possibility of having a behavior of reciprocity towards these nodes.

We plan the simulation of an extension to OLSR using trust rules for MPR selection and routing table calculation. Another possibility is to set up a trust management module to be tied to the structure of the nodes without modifying the protocol. Our goal is to measure the impact of these solutions on the protocol, while preserving the auto-organization and the dynamic of the adhoc environment. With regard to the usage of an explicit specification of direct and derived trust relations, it is worth, in the view of trust, to compare OLSR with other protocols, for example AODV, and report the contribution of trust to the security of both protocols.

Acknowledgements. Rafael Timóteo de Sousa Jr. is supported by CNPq - Brazil and would like to thank Supélec - France for its support during his postdoctoral program.

References

1. Mui L (2003) Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks, PhD Thesis, Massachusetts Institute of Technology.
2. Clausen T, Laouiti A, Muhlethaler P, Raffo D, Adjih C (2005) Securing the OLSR routing protocol with or without compromised nodes in the network, HAL - CCSd - CNRS, INRIA - Rocquencourt.
3. Clausen T, Jacquet P (2003) IETF RFC-3626: Optimized Link State Routing Protocol OLSR.
4. Yahalom R, Klein B, Beth T (1993) Trust Relationships in Secure Systems - A Distributed Authentication Perspective. In: SP'93: Proceedings of the 1993 IEEE Symposium on Security and Privacy. IEEE Computer Society, Washington, USA.
5. Marsh S (1994) Formalising Trust as a Computational Concept, PhD Thesis. Department of Mathematics and Computer Science, University of Stirling.
6. Raffo D (2005) Security Schemes for the OLSR Protocol for Ad Hoc Networks, PhD Thesis, University of Paris 6 *Pierre et Marie Curie.*
7. Buchegger S (2004) Coping with Misbehavior in Mobile Ad-hoc Networks, PhD Thesis. IC School of Computer and Communication Sciences, Lausanne university.
8. Fourati A, Al Agha K (2006) A Shared Secret-based Algorithm for Securing the OLSR Routing Protocol. Springer Netherlands, Telecommunication Systems, Volume 31, Numbers 2-3, pp. 213-226.
9. Jensen C D, Connell P O (2006) Trust-Based Route Selection in Dynamic Source Routing. In: Trust Management, 4th International Conference, iTrust 2006. Springer, Volume 3986/2006, pp.150-163, Pisa, Italy.
10. Michiardi P (2004) Cooperation Enforcement and Network Security Mechanisms for Mobile Ad Hoc Networks, PhD Thesis, Ecole nationale supérieure des télécommunications, Paris.
11. Tanachaiwiwat S, Dave P, Bhindwale R, Helmy A (2004) Location-centric Isolation of Misbehavior and Trust Routing in Energy-constrained Sensor Networks. IEEE International Performance, Computing, and Communications Conference (IPCCC), pp. 463-469.
12. Liu J, Issarny V (2004) Enhanced Reputation Mechanism for Mobile Ad Hoc Networks. In: Trust Management. 2nd International Conference, iTrust 2004. Springer, Volume 2995/2004, pp. 48-62, Oxford, UK.
13. Grandison T, Sloman M (2000) A Survey of Trust in Internet Applications. IEEE Communications Surveys and Tutorials, 4th Quarter, Vol. 3, No. 4.
14. Viljanen L (2005) Towards an Ontology of Trust. In: Trust, Privacy and Security in Digital Business. Springer, Volume 3592/2005, pp. 175-184.
15. Puttini R S, Mé L, Sousa Jr R T (2004) On the Vulnerabilities and Protection of Mobile Ad Hoc Network Routing Protocols. In: Proceedings of the 3rd International Conference on Networking ICN'2004. IEEE, pp. 676-684, New Jersey, USA.
16. Nanz S, Hankin C (2006) A Framework for Security Analysis of Mobile Wireless Networks. Theoretical Computer Science, Volume 367, pp. 203–227.
17. Bussard L (2004) Trust Establishment Protocols for Communicating Devices, PhD Thesis, Eurecom - ENST.
18. Johnson D B (1973) A Note on Dijkstra's Shortest Path Algorithm, Journal of the ACM, Volume 20, pp. 385-388, New York, USA.