João Paulo Vilela

Reputation-based Security for Optimized Link State Routing in Wireless Ad-hoc Networks

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
2006

Luís Antunes

João Paulo Vilela

Reputation-based Security for Optimized Link State Routing in Wireless Ad-hoc Networks

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
2006

João Paulo Vilela

Reputation-based Security for Optimized Link State Routing in Wireless Ad-hoc Networks

Tese submetida à Faculdade de Ciências da Universidade do Porto para obtenção do grau de Mestre em Ciência de Computadores

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
2006

Aos meus pais.

Acknowledgments

I would like to begin by thanking my supervisor, João Barros. The first time I talked to him I was finishing my undergraduate studies and trying to decide what to do next. I went to do an internship in a private company, after which, confined by his positive and dynamic attitude, I decided to come back to faculty to work with him. Since then, João was able to create a research group which truly reflects my ideals of what a research group should be and that includes, among other things, people regularly discussing their work with each other, and regularly having exciting people visiting us to talk about their work and hear about ours. I would like to express my gratitude for his constant support, encouragement and guidance. I would also like to thank him the revisions and suggestions towards the improvement of this thesis.

My excitement about our research group is also due to the keen environment I have been sharing on a daily basis with my room colleges Gerhard Maierbacher, Luísa Lima, Paulo Martins, Paulo Falcão e Rui Costa, and my "neighbors" David Pereira and Jorge Coelho. A few words to them to state how much I appreciate their regular company. Our group would not be complete without the experience, readiness and thoughtful attitude of Pedro Brandão, Rui Prior and Sérgio Crisóstomo. I cannot end up without mentioning Susana Amorim for always receiving us with a smile even when she is full of things to do.

To my close friends and all of those which, at some point in my life contributed to shape my personality and help me becoming who I am today, thank you all!

My life would not be complete if, apart from doing something I enjoy at a professional level, I would not have Tânia in it. It is a great motivation to know that a day of work will end with her company, independently on how everything else flows in my life.

I do not have words to express my gratitude towards my parents for always showing interest, enthusiasm and giving support for every decision I make, and for providing me with all the conditions to do my best from early age. This work is dedicated to them.

Abstract

A Mobile Ad-hoc Network (MANET) is a dynamic self-organizing collection of devices communicating over the wireless medium. In such networks no central administration is used, instead each network node must be able to detect and establish routes through other nodes in the network. Thus, MANETs are infrastructureless networks which do not require any fixed infrastructure for their operation. This kind of networks is also referred to as multi-hop ad-hoc networks because typically they depend on having network nodes relaying traffic on behalf of other nodes in the network through paths with more than a single hop.

A great number of routing protocols for MANETs have been proposed in the last few years which, due to the distributed nature of this networks, rely on the information provided by network nodes. In environments such as disaster relief, it is reasonable to assume network devices to be highly cooperative, contributing to the network operation. In other environments, with devices constrained by low power and low battery capabilities, it is likely that users restrain from cooperating in the network operation for their own gain, resulting in a severe degradation of the network performance.

Therefore, mechanisms to enforce user cooperation by encouraging compliance with the rules of the routing protocol assume crucial importance. The cooperation enforcement mechanisms proposed so far are either currency-based mechanisms, where a virtual currency is used as payment of routing operations, or reputation-based mechanisms, where network nodes attempt to detect and isolate or punish misbehaving nodes.

In this thesis we present and evaluate, both analytically and through simulation, a reputation-based security scheme which deals with the generation of fake routing control traffic in MANETs. Moreover, we describe in detail the issues that arose during the implementation and validation process of the protocol, this way illuminating the difficulties in bringing reputation-based concepts to a real-world environment. We also propose solutions to address those identified issues.

Resumo

As redes móveis ad-hoc (tipicamente designadas MANETs) são redes dinâmicas e auto-organizáveis constituídas por dispositivos que comunicam através de um meio sem fios. Ao invés do paradigma mais comum, a manutenção deste tipo de redes é efectuada de forma completamente distribuída. Os dispositivos da rede são responsáveis por detectar e determinar caminhos através dos dispositivos vizinhos por forma a conseguir comunicar com todos os dispositivos que compõem a rede.

Recentemente, vários protocolos de encaminhamento para as MANETs foram propostos cuja operação depende, devido à natureza descentralizada das MANETs, de forma crucial da informação disponibilizada pelos dispositivos da rede. Se em determinados ambientes, como por exemplo auxílio a situações de catástrofe, é natural assumir uma elevada cooperação na manutenção da rede, outros há em que é espectável que os utilizadores se retraiam em participar na mesma (por terem, por exemplo, dispositivos com bateria limitada), o que resulta numa diminuição elevada da capacidade da rede.

Neste sentido, a existência de mecanismos para estimular a cooperação nas tarefas de manutenção da rede é crítica para o eficaz funcionamento da mesma. Os mecanismos de estimulo à cooperação actuais são de dois tipos: mecanismos "monetários", baseados na troca de dinheiro virtual como forma de pagamento pela participação na manutenção da rede, e mecanismos baseados em reputação, nos quais os utilizadores tentam detectar e castigar ou isolar utilizadores com mau comportamento.

Nesta tese apresentamos e avaliamos, por meios analíticos e por simulação, um mecanismo baseado em reputação para estimular a correcta geração de mensagens de controlo fulcrais ao funcionamento dos protocolos de encaminhamento. Adicionalmente, descrevemos os problemas encontrados durante a implementação e validação do nosso sistema, desta forma ilustrando as dificuldades inerentes à utilização de sistemas de reputação em ambientes reais. Propomos também soluções para obviar essas dificuldades.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A Mobile Ad-hoc Network (MANET) is a dynamic self-organizing collection of devices communicating over the wireless medium. From now on we will refer to these devices as network nodes or simply nodes. In such networks no central administration is used. Instead, each node must be able to detect the presence of other nodes, establish the corresponding connections and determine how to reach other nodes in the network. Thus, MANETs are infrastructureless networks since they do not require any fixed infrastructure such as base stations, neither any specialized entities such as routers for their operation. Therefore, crucial tasks for the management of this kind of networks is performed with the contribution of every node that comprises it, in a fully distributed manner.

The nodes of a MANET are free to move around while communicating with other nodes and, therefore, the network topology can change rapidly and in unpredictable ways. These networks are typically based on having nodes relaying other nodes traffic through paths with more than a single hop, thus being also designated as multi-hop ad-hoc networks.

Historically, MANETs have been envisioned for military applications, e.g. communication among a group of soldiers for tactical operations, and placement in inhospitable terrains.

With the rapid advances of research in mobile ad-hoc networking in the last few years, other kinds of application interests have gathered around ad-hoc networks. These include home networks, e.g. a user device communicating with home wireless devices to adjust environmental settings to the user preferences, open doors, activate and deactivate lights, etc. Two other applications which are now attracting growing atten-

tion are wireless mesh networks and vehicular communications. The first can be used to extend the coverage of current networks with reduced infrastructure requirements, while the usage of the second can contribute to improve road safety and optimize road traffic.

Many other applications of ad-hoc networks exist with specific requirements and constraints such as sensor networks and personal area networks.

Characteristics of MANETs

Due to the dynamic and distributed nature of this kind of networks and to the specific characteristics of the wireless medium, several things have to be taken into consideration when developing protocols for this kind of networks. E.g. the same medium is shared by multiple mobile ad-hoc nodes which may or may not be visible by every node interested in the communication. The mobility of network nodes results in frequent topology changes with links arising and breaking very often. With devices and their batteries becoming smaller, power consumption limitations also have to be considered. Since there is no centralized management, the successful operation of such networks requires a minimum amount of cooperation between nodes in the network. This requirement is particularly prominent with respect to the discovery and establishment of routes for reliable data delivery.

1.1 Daidalos Framework

The work of this thesis was carried out as part of the EU-funded project Daidalos II. Daidalos stands for Designing Advanced network Interfaces for the Delivery and Administration of Location independent Optimized personal Services. Daidalos II is an Integrated Project (IP) of the EU *Framework Programme 6.*

The aim of Daidalos II is to radically improve user-friendliness and economic viability in the European information and communication sector by integrating mobile and broadcast communications and following a user-centered, scenario-based and operator-driven approach to deliver services pervasively and seamlessly across heterogeneous networks.

In terms of the overall working structure, Daidalos II is composed by 5 work-packages (WPs). WP1 is responsible for the development of scenarios, the overall framework

and architecture and for defining methodologies. The implementation WPs, WP2-4, are responsible for the detailed technical work, whereas WP5 is responsible for system integration, the validation platform and overall testing.

The work on this thesis is related to the WP2. The goal of WP2 is achieving an efficient and scalable integration of heterogeneous access network technologies, including cellular, satellite, broadcast, wired and wireless networks both infrastructure-based (e.g. Wireless Local-Area Network (WLAN), Wireless Metropolitan Access Network (WMAN)) and infrastructureless (e.g. ad-hoc networks, personal networks and sensor networks).

Within the scope of the project, ad-hoc networks are considered as an alternative way for the user terminal to join a network and access services through an infrastructure, operator managed, network. Security issues of this class of networks, with special emphasis on security concepts at the network, have to be addressed. This is where our work comes to place, namely at the level of secure routing for ad-hoc networks.

Since Daidalos II is a user-centered project, it makes sense to consider the security of ad-hoc networks based on identities (e.g. user, group or network operator). Thus, a close interaction with the security and privacy infrastructure of WP3 and WP4, which provides means for Authentication, Authorization, Accounting, Auditing and Charging (the so called *A4C*), was taken into account.

The ad-hoc network concept of Daidalos II is an extension to the infrastructure network and, therefore, there exists a point of access to the infrastructure network which provides, among other things, the aforementioned security and privacy infrastructure. The ad-hoc network as seen from the project is a variant of a pure ad-hoc network, which provides more functionalities than those expected of a pure ad-hoc network where every task is performed in a fully distributed manner. Although maintaining the project perspective in mind, for the sake of generality we have developed our work around the concept of a pure ad-hoc network without any centralized entity. This work is easily adopted to the project perspective of an ad-hoc network because the later provides more functionalities than the pure ad-hoc network considered.

1.2 Main Contributions

Our main contributions are as follows.

- We provide a taxonomy of security vulnerabilities specific of the Optimized Link State Routing (OLSR) protocol for MANETs;
- We present a reputation-based security scheme to address the generation of fake routing protocol control traffic;
- We perform an analytical evaluation of the overhead induced by the aforementioned reputation-based security scheme;
- We present the results of a simulation-based analysis performed to underline the effectiveness of our reputation-based security scheme;
- We provide a thorough description the issues that arose during the implementation and validation process of the reputation-based security scheme, and propose mechanisms to tackle them.

1.3 Outline of the Thesis

The outline of the thesis is the following. In Chapter 2 we provide a general description of routing protocols for Mobile Ad-hoc Networks, the current solutions to secure them in terms of the inherent cooperation aspects of this kind of networks. Afterwards, in Chapter 3, the OLSR protocol is described, a taxonomy of its security vulnerabilities is catered, and the previous work on security is outlined and analyzed. Chapter 4 comprises a thorough description of our security scheme for OLSR followed by a description and discussion of the performance evaluation of it on Chapter 5. Chapter 6 concludes the thesis.

Chapter 2

Secure Routing

In this chapter we provide a review of the current solutions for secure routing in MANETs. Many of the security schemes proposed work as extensions to the already available routing protocols, therefore we start by describing the standard routing protocols for MANETs. Afterwards, we follow through the current secure routing solutions for MANETs, with special emphasis on the aspects of cooperation enforcement.

2.1 Routing in Mobile Ad-hoc Networks

A great number of routing protocols for MANETs have been proposed in the last few years and they can be classified from very different perspectives [MM04, BCGS04] according to their different nature. In this section we will divide routing protocols into proactive or reactive, depending on whether they maintain routing tables that are readily available or discover routing paths on demand.

2.1.1 Proactive/Table-driven

A proactive or table-driven routing protocol maintains topology information of the network obtained from a periodic exchange of control traffic containing such information. The exchange of routing information is performed regardless of whether route calculations are needed.

Destination-Sequenced Distance Vector (DSDV)

The DSDV protocol [PB94] is an extension of the distance vector protocol concept for MANETs. It is based on having each node periodically send all or part of their routing tables to all nodes in the network. The tables are exchanged at regular intervals between neighbor nodes and, if significant changes in the topology occur, they are also forwarded (either entirely or partially) to other nodes. Using the information from the tables, each node updates its own routing table based on the received information and the metric considered (typically the number of hops).

To tackle the count-to-infinity problem, prevent loops and speed up the convergence, the update of tables within DSDV comprises increasing sequence numbers.

Optimized Link State Routing (OLSR)

The OLSR protocol [JMC+01] is a link state routing protocol composed by two main operation mechanisms: a neighbor discovery mechanism and a mechanism for dissemination of topology information.

The neighbor detection is based on a periodic exchange of HELLO messages between neighboring nodes with the intent of providing information of the neighbors up to two hops away. The HELLO messages contain a list of neighbors and the corresponding link state for each of them. The dissemination of topology information is performed through the diffusion of TC (Topology Control) messages through the network, announcing a list of specific neighbors called the *multipoint relay selector set*.

The key toward the efficiency of the OLSR protocol is an ingenious link state packet forwarding mechanism, based on having each node select a subset of its neighbors such that this subset ensures connectivity to every two-hop neighbors. The nodes on this subset are called *multipoint relays* (MPRs) and the subset is the *multipoint relay set* (MPR set). The *multipoint relay selector set* mentioned above is the set of nodes which select a certain node as a MPR.

The use of MPRs for control traffic transmission results in a scoped flooding instead of a full node-to-node flooding thus inducing a reduction on the amount and size of exchanged control traffic.

Topology dissemination Based on Reverse-Path Forwarding (TBRPF)

The TBRPF protocol [OTL04] is also a link state routing protocol. As with OLSR, it consists of two main mechanisms: a neighbor discovery mechanism and a mechanism for topology information dissemination.

The neighbor discovery mechanism is also based on a periodic exchange of HELLO messages to provide information about each node to neighbor nodes.

The mechanism for topology information dissemination is based on each node periodically sending all or a part of a shortest-path source tree (calculated through a modified version of the Dijkstra's algorithm) to each node in the network. Two types of control messages are used to report either the full tree or a partial tree called *reportable subtree.*

2.1.2 Reactive/On-demand

A reactive or on-demand routing protocol does not maintain topology information of the network. Instead, whenever a route to a certain destination is needed, it generates a request for the desired path.

Ad hoc On-Demand Distance Vector Routing (AODV)

AODV [PR99] is an on-demand routing protocol in which the routing discovery process is based on broadcasting messages (designated RREQ) requesting the path to a destination when one is needed, and receiving unicast messages (designated RREP) providing the requested path if such exists. One additional type of message, designated RERR, is used to announce link breakages.

AODV uses sequence numbers to avoid loops and guarantee the selection of the most recent routes. The information provided by the control messages of AODV is processed by both the end nodes and the intermediate nodes, which store next-hop information for each flow for data packet transmission. Intermediate nodes also have the ability to respond to route requests if they have a fresh route to the destination.

Dynamic Source Routing (DSR)

DSR [JM96] is an on-demand source routing protocol which uses the same type of control messages as AODV: RREQ for requesting routes, RREP for replies and RERR

to announce link breakages. Since it is a source routing protocol, each data packet contains a strict source route which specifies the path it shall take in the network.

To determine the path to a certain destination the source node broadcasts a RREQ message to the network. When a neighbor node receives the RREQ it updates its route to the source, adds its IP address to the RREQ path and forwards the message. As the message traverses the network each node's IP address accumulates in the RREQ path and, when this message is received by intermediate nodes, they can update the routes for each node contained in the corresponding path. If an intermediate node is aware of the remaining path he appends it to the current path in the RREQ and sends a RREP to the source or forwards the message as usual. When the message reaches the destination, a RREP is sent back with the full path to the destination. When a link break occurs in a determined path, a RERR is sent to the source by a node subsequent to the break point.

DSR holds routing information in a route cache instead of a routing table which allows him to retain several possible paths to a certain destination.

2.2 Cooperation Aspects

Due to the distributed nature of this type of networks, the proposed MANET routing protocols rely on the information provided by network nodes. In their original specification, most of the protocols consider nodes in the network as benign in the exchange of crucial routing information. However, the issues of selfish behavior and disruption of the routing protocol operation cannot be discarded since they may cause a severe degradation in network performance [MM02b].

Several proposals to secure routing protocols exist. Many of them are cryptography-based solutions, but some offer cooperation enforcement solutions [BCGS04, ID03]. In this section we will provide an overview of the latter since they are of utmost relevance for our work.

Cooperation enforcement mechanisms can be divided in two categories: currency-based mechanisms and reputation-based mechanisms.

The currency mechanisms are based on exchange of virtual currency between nodes in the network [BH00] or on the availability of a service which trades credits by receipts retrieved from messages in transit in the network [ZCY03].

In terms of reputation-based solutions, they are typically composed by three distinct mechanisms: (1) a local monitoring mechanism to observe the behavior of network nodes and determine their trustworthiness, (2) a reputation dissemination mechanism to convey other nodes with the results from the observations performed by the previous mechanism, and (3) a punishment/isolation mechanism to protect the network from malicious behavior.

2.2.1 Nuglets

Nuglets are a virtual currency used to pay for packet forwarding services [BH00]. Two operational models have been defined: the Packet Purse Model and the Packet Trade Model. In the Packet Purse Model, the source node loads nuglets in the packet before sending it and each intermediate node acquires some of this nuglets as payment when they forward it. If the nuglets run out before reaching the destination, the packet is dropped. In the Packet Trade Model nuglets are exchanged by nodes in the path, instead of being kept in the packets. Each forwarding node buys the packet from the previous node by some nuglets and sells it to the following node for more nuglets, this way earning nuglets. The total cost of the operation is supported by the destination node.

Both of the approaches rely on a tamper proof security module. The authors recognize that it is difficult to estimate the number of nuglets needed to send in the packet in the Packet Purse Model, and the Packet Trade Model allows overloading of the network because the sources are not bound to pay for sending packets.

The mechanism proposed in [BH03] overcomes the issue of the estimation of the amount of nuglets to send, because nuglets are not carried in packets. Instead, a counting technique is used, where each node holds a nuglet counter which is decreased when a node sends an own packet and increased when he forwards packets on behalf of other nodes.

2.2.2 Watchdog and Pathrater

The watchdog and pathrater [MGLB00] are two extensions to the DSR routing protocol that attempt to detect and mitigate the effects of routing misbehavior.

The watchdog is a mechanism for detection of misbehavior based promiscuously monitoring the next node in the path to detect if he correctly forwards packets sent to it.

If a node bound to forward a packet fails to do so after a certain period of time, the watchdog decrements a failure rating for that specific node. A node is considered as misbehaving when this failure rating exceeds a certain threshold.

The pathrater uses the information gathered by the watchdog to determine the best possible routes. This is done by calculating a metric for each path based on the reliability of the nodes contained in the path and subsequently selecting the highest rating paths for communication.

This mechanism does not punish misbehaving nodes (it actually relieves them from forwarding operations), instead it facilitates the avoidance of misbehaving nodes thus resulting in a selection of better paths and the consequent increase in throughput in a network with malicious nodes.

2.2.3 CONFIDANT

CONFIDANT stands for Cooperation Of Nodes, Fairness in Dynamic Ad-hoc NeTworks [BB02]. It is an extension to DSR composed by the following components: a monitor for detection of deviating nodes, a trust manager to determine the trustworthiness of received alarm messages, a reputation system which holds reputation information obtained through observed or reported behavior, and a path manager to isolate malicious nodes and participate in the path ranking and selection.

The monitor mechanism detects deviations by listening to the transmissions of the next node in the path to detect relay refusal attacks. Other kind of deviations can be detected by observing routing protocol behavior or by keeping a copy of sent packets to detect content changes.

The trust manager is responsible for sending and receiving alarm messages, and for managing the trust given to received alarms according to the trust levels of the source nodes.

The reputation system manages the ratings of nodes in the network. These ratings are changed according to a rate function which assigns different weights to different types of malicious behavior detections, namely giving prevalence to local observations over second-hand information provided by other nodes.

The path manager participates in the route selection mechanism by deleting routes that contain nodes which have been classified with an intolerable rating, and performs actions with the intent of isolating malicious nodes (e.g. ignore route requests from

malicious nodes, alert sources of route requests which hold a malicious node in the source route).

This protocol is subject to spreading of wrong accusations, which was addressed by the authors through the use of Bayesian statistics for classification and exclusion of liars.

2.2.4 CORE

CORE is a COllaborative REputation mechanism to enforce node cooperation in MANETs. It is a general scheme composed by a validation mechanism with respect to a certain defined function and three types of reputation (subjective, indirect and functional) which are combined in a global reputation value.

Through the validation mechanism each node monitors the behavior of its neighbors by collecting observations about the execution of some mechanism. The example provided is the previously described watchdog mechanism applied to the DSR routing protocol to detect misbehaving nodes. When a node generates a Route Request it monitors subsequent nodes verifying if they follow the expected behavior, either by sending a Route Reply or relaying the Route Request. If the neighbor follows the protocol rules, the node observation is positive and negative otherwise.

This validation mechanism is complemented by a sophisticated reputation mechanism that considers three reputation types. The subjective reputation is based on performed observations and avoids sporadic misbehavior by giving relevance to past observations in its calculation. The indirect reputation reflects the possibility of interactions between non-neighbor nodes by considering solely positive information (to avoid denial of service attacks) provided by other nodes in the network. The functional reputation refers to a reputation based on the observation of different operational functions (e.g. routing and packet forwarding) combined in a global reputation value.

To enforce the cooperation of nodes, the execution of a function requested by a node in the network is conditioned by the corresponding reputation value. Misbehaving nodes may be reintegrated into the network if they increase their reputation by cooperating in the network operation.

2.3 Summary

Several routing protocols have been proposed for ad-hoc networks which rely on the assumption that nodes are benign in the exchange of crucial routing information. Although this is reasonable in some scenarios of application, such as disaster relief, there are others were the owners of the devices are likely prone to avoid spending their constrained device capabilities on behalf of other users. Details studies show that a small amount of non-cooperating users have the ability to seriously degrade the network operation.

Therefore, apart from the cryptographic security solutions to guarantee authentication and integrity within the network, it is essential to have mechanisms to enforce user cooperation by providing incentives to cooperate and/or punishing cooperation refusal. The solutions developed so far are basically of two kinds: currency-based solutions which depend on components which may reduce their widespread applicability, and reputation-based solutions which rely on the ability to securely identify nodes in the network.

To the best of our knowledge, reputation-based security schemes for proactive routing protocols such as OLSR have not yet been implemented.

Chapter 3

Security Aspects of the OLSR Protocol

In this chapter we start by describing the Optimized Link State Routing (OLSR) protocol, proposed by Jacquet et. al in 2001, with sufficient detail to enable a thorough understating of the security issues subsequently presented, and the currently proposed extensions to secure the protocol.

3.1 Protocol Description

The OLSR protocol can be classified as a proactive or table-driven link state routing protocol. As a proactive routing protocol it regularly exchanges topology information among network nodes, which results in the advantage of making the routes immediately available when needed. As a link state protocol, it uses flooded information about the network topology to build a map of the network and then calculate the best next-hop for every possible destination.

OLSR offers, in fact, more than a pure link state protocol, because it provides the following features:

- *minimization of flooding* by using only a set of selected nodes, called *multipoint relays* (MPRs), to diffuse its messages to the network;
- *reduction of the size of control packets* by declaring only a subset of links with its neighbors who are its *multipoint relay selectors* (MPR selectors).

3.1.1 Multipoint Relays

The OLSR protocol is an optimization of the classical link state algorithm based on a mechanism called *multipoint relay* (MPR). The intent of multipoint relays is to minimize the flooding of the network with broadcasted packets by reducing duplicate retransmissions in the same region. Each node selects a set of its neighbor nodes that will retransmit its packets. This set of nodes is called the *multipoint relay set* (MPR set) of that node and only the nodes in the MPR set are responsible for forwarding control traffic intended for diffusion into the entire network. The node which chooses the multipoint relay set is a *multipoint relay selector* (MPR selector) for each node in the set.

Each node selects its MPR set in a way such that it contains a subset of one-hop neighbors covering all the two-hop neighbors. Additionally, all two-hop neighbors must have a bi-directional link to the selected MPR set. The smaller the MPR set, the more efficient the routing protocol. For details about the computation of the MPR set see [CJ03].

OLSR determines the routes to all destinations through these nodes, i.e. MPR nodes are selected as intermediate nodes in the path. The scheme is implemented by having each node periodically broadcast control traffic information about the one-hop neighbors that selected it as a MPR (or, equivalently, its MPR selectors). Upon receiving information about the MPR selectors, each node calculates and updates its routes to each known destination. Consequently, the route is a sequence of hops through MPRs from the source to the destination. The neighbors of any node which are not in its MPR set receive and process the control traffic but do not retransmit it.

The use of MPRs for message transmission results in a scoped flooding instead of full node-to-node flooding (see also Fig. 3.1, where the nodes in thick blue are the MPRs of node 1), thus inducing a reduction of the amount of exchanged control traffic. The protocol is particularly well suited for large and dense networks, because the optimization procedure based on MPRs works best in those cases.

3.1.2 OLSR Control Traffic

There are two main types of control messages in OLSR: HELLO and TC (Topology Control) messages.

1) HELLO messages are periodically broadcasted by each node, containing its own

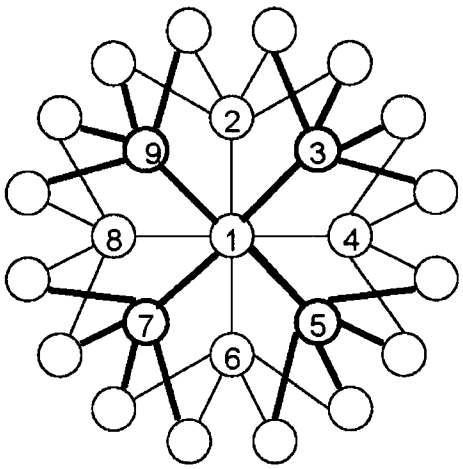

Figure 3.1: Multipoint relay optimization in action

address and three lists: a list of neighbors from which control traffic has been heard but no bi-directionality has been confirmed, a list of neighbors with which bi-directionality has already been confirmed, and the list of neighbors which have been selected to act as MPRs for the originator node. These messages are only exchanged between neighboring nodes but they allow each node to have information about one and two-hop neighbors which is later used in the selection of the MPR set;

2) TC messages are also emitted periodically by nodes in the network. These messages are used for diffusing topological information to the entire network. A TC message contains the list of neighbors who have selected the sender node as a MPR (MPR selector set) and a sequence number associated to the MPR selector set. This information is later used in route calculations.

Each OLSR control message can be uniquely identified through a tuple consisting of its *originator address* and its *message sequence number*. A node may receive the same message several times, therefore, to avoid duplicate processing and transmission of control traffic, each node maintains a *duplicate set* where the unique identifier of each message is stored, along with a boolean value that indicates whether the message has already been transmitted. This mechanism is called the duplicate transmissions avoidance mechanism.

Through the exchange of OLSR control messages each node stores the following information about the network. The available links to neighbor nodes are kept in the *link set*, the neighbor nodes themselves are kept in four sets according to their nature: the one-hop neighbors are kept in the *neighbor set*, the two-hop neighbors and

the nodes which provide access to them in the *neighbor 2-hop set*, the chosen MPRs in the *MPR set* and the nodes which selected it as MPR in the *MPR selector set*. Nodes also keep information about the network topology gathered from TC messages which is stored in the *topology set* in the form of tuples consisting mainly of a destination address and a last-hop address to that destination.

3.1.3 Protocol Operation

In summary, the OLSR protocol can be specified as follows.

1. Each node periodically broadcasts its HELLO messages;
2. These are received by all one-hop neighbors but are not relayed;
3. HELLO messages provide each node with knowledge about one and two-hop neighbors;
4. Using the information from HELLOs each node performs the selection of their MPR set;
5. The selected MPRs are declared in subsequent HELLO messages;
6. Using this information each node can construct its MPR selector table, with the nodes that selected it as a multipoint relay;
7. A TC message is sent periodically by each node and flooded in the network, declaring its MPR selector set;
8. Using the information of the various TC messages received, each node maintains a topology table which consists of entries with an address of a possible destination (a MPR selector in the TC message), an address of a last-hop node to that destination (the originator of the TC message) and a MPR selector set sequence number;
9. The topology table is then used by the routing table calculation algorithm to calculate the routing table at each node. Details about this procedure may be found in [JMC+01] and [CJ03].

3.2 Taxonomy of Security Vulnerabilities in OLSR

In its original specification, the OLSR protocol has the underlying assumption that all nodes are benign in the exchange of crucial topology information through control traffic, which makes it vulnerable to several attacks.

In a proactive routing protocol, each node has two tasks to accomplish [ACJ+03]: (1) correctly generate the routing protocol control traffic (this way giving correct information to the other nodes on the network) and (2) correctly relay the routing protocol traffic on behalf of other nodes (this way allowing for the control traffic to reach every node in the network). Thus, an attack on the routing protocol must result as the corruption of one of this tasks by some node. This can be accomplished by four main actions:

1. *Fabrication of false routing messages:* A node generates regular routing control traffic messages containing false information or omitting information of the current state of the network;
2. *Refuse of control traffic generation/relay:* A node refuses to generate its own routing control traffic or refuses to forward other nodes control traffic;
3. *Modification of routing control traffic:* A node does relay other nodes' traffic but modifies it to insert wrong information or omit information from the network;
4. *Replay attacks:* A node listens to routing control traffic transmissions on the network and later on injects possibly wrong and outdated information in the network.

Table 3.1 gives a taxonomy of OLSR security vulnerabilities and provides examples of attack actions based on the network illustrated in Fig. 3.2.

We do not consider the *jamming attack* in which an attacker saturates the medium by sending a large amount of messages, reducing the other nodes ability to communicate, because it results from the inherent characteristics of the communication medium and is independent of the routing protocol employed.

3.3 Previous Work on OLSR Security

Recently, several contributions have appeared, aimed at securing OLSR [ACJ+03, ARM05, ACL+05, RACM04]. In the following, we provide an overview of their main features, identifying the underlying assumptions and unsolved issues.

The proposal in [ACJ+03] is based on a mechanism for key distribution and establishes a line of defense in which (i) nodes are either trusted or untrusted and (ii) trusted

ATTACK	METHOD	EXAMPLE	TARGET	RESULT
Identity spoofing	Fake HELLO	M_3 generates HELLOs pretending to be A	All nodes	MPR nodes of M_3 will present themselves as last-hop for node A, resulting in conflicting routes to node A.
Link spoofing	Fake HELLO	M_1 generates HELLOs advertising bi-directional links to most of A's two-hop neighbors	Specific node	A chooses M_1 as its main MPR[4] which allows M_1 to intercept and modify most of A's traffic
	Fake TC	M_1 generates TCs advertising D as his MPR selector, directly to G[5]	Group of nodes	Distance between M_1 and D will be deemed to be one hop, thus M_1 will become the main bridge between G and D
	Routing table overflow	M_1 generates many TCs containing non-existing nodes in the MPR set[6]	All nodes	The routing table algorithm will lose a lot of time calculating false routes
Traffic relay/ generation refusal	Drop packets/ Blackhole	After becoming a preferential relay choice for A or G[7], M_1 drops packets received from them	Specific node Group of nodes	Loss of connectivity / Degradation of communications
	Refuse to generate control traffic	M_1 is selected as MPR for A and does not advertise that information to the network	Specific node	Node A unreachable, degradation of communications
Replay attacks	Traffic replay	M_1 sends to other nodes "old" previously transmitted[8] TC or HELLO messages	All kinds	Outdated, conflicting and/or wrong information enters the network which may cause defective routing
Wormhole	Protocol disobedience	M_2 and M_3 collude and exchange packets encapsulated, without the modifications presumed by the routing protocol	All kinds	The extraneous inexistent link M_2 - M_3 becomes a preferential choice for traffic and is fully controlled by M_2 and M_3

Table 3.1: Taxonomy of OLSR security vulnerabilities

Examples presented are based on Fig. 3.2. ($M_{1,2,3}$ - malicious nodes, A - attacked node, D - destination node, G - group of nodes). [4] Because the smaller the MPR set is, the more efficient the OLSR results are; [5] M_1 is one hop away from G nodes; [6] I.e. declaring non-existing nodes and links; [7] It may use e.g. the described link spoofing techniques; [8] The messages can also be correctly authenticated.

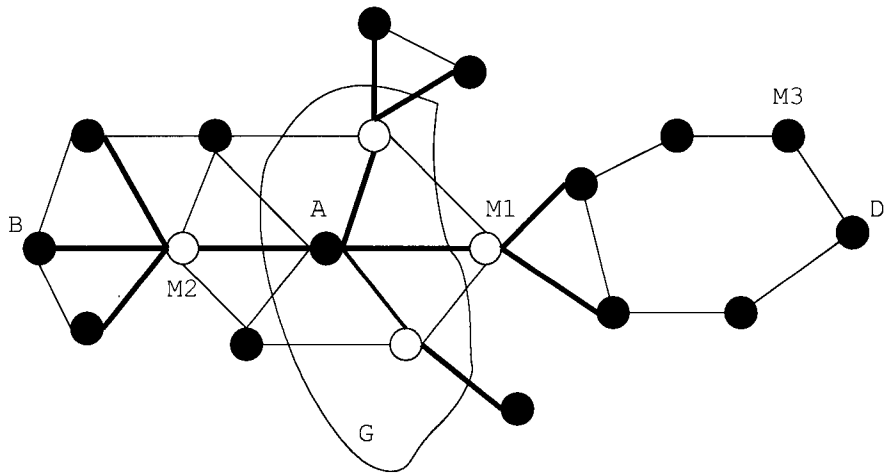

Figure 3.2: Exemplary network topology for the OLSR protocol.

Nodes in gray are MPRs of node A; light edges represent the connections between nodes; dark edges identify the used links between A and all of its two-hop neighbors through the selected multipoint relay set. $M_{1,2,3}$ denotes the malicious nodes, D is the destination node and G defines a group of nodes.

nodes are not compromised. It entails a timestamp and a signature associated with each routing control message: the signature is used to identify messages from trusted nodes, and timestamps are used to prevent the replay of old messages. The approach does not contemplate the following issues: (a) trusted nodes may behave incorrectly because of malfunctioning, unconsciously corrupting the routing protocol; (b) nodes in MANETs typically get in and out very often, thus it is hard to separate nodes into trusted and untrusted; (c) the signing mechanism is not detailed (a possibility is [RACM04]).

The contribution in [RACM04, Raf05] considers the compromise of trusted nodes. It is assumed that a public key infrastructure (PKI) and a timestamp algorithm (e.g. the one in [ACJ+03]) are in place. An additional message (ADVSIG) is sent in conjunction with routing control traffic. This message contains timestamp and signature information. Each node has a so called *Certiproof* table where information received in ADVSIGs is kept. This information is then reused as a proof of correctness of the link state information in subsequent messages. The procedure ensures that a lone attacker node is not able to send wrong link state information to the network. Its drawbacks are as follows: (a) it does not protect against denial of service or wormhole attacks (see Section 3.2) and (b) it imposes a large overhead to the network in terms of additional traffic and computation of signatures.

The focus of [ARM05] is on distributed key management techniques, providing a brief overview of methods to prevent wormhole and message replay attacks (see also Section 3.2). The technique to prevent wormhole attacks is based on a variant of the counting technique [ACL+05] in which nodes advertise a set of hashes of the packets received over each of the last k intervals. This way it is possible to check if packet losses cross a certain threshold, in which case a node is assumed to be compromised. Replay attacks are prevented by the use of timestamps.

The security mechanism proposed in [ACL+05] uses signature and timestamp schemes to ensure authentication and protection against replay attacks. The signature technique is based on sending a signature with each routing control message as in [RACM04]. Also proposed is a scheme to counter relay attacks based on the geographical position of nodes and a scheme that deals with compromised nodes based on *network flow conservation*, where misbehavior in traffic relaying is detected based upon the number of packets sent and received by each node. The drawbacks of this proposal are as follows: (a) the weak assumption that forwarding the correct number of packets by a node proves that the packets were sent properly; and (b) a centralized security authority that manages misbehavior detection and remedy is difficult, if not impossible, to implement in a MANET.

In [DRWL04] a fully distributed certificate authority (DCA) based on threshold cryptography is described. The CA is distributed in the way that a node requests a certificate from any coalition of k nodes (shareholders) of the network. Upon the certificate request, each of the shareholders determines if he wants to serve the request based on whether the requesting node is well behaving. Upon receiving k "partial certificates" they are manipulated to generate a valid certificate as if it was signed by a regular CA. A monitoring system used to determine behavior of network nodes is not incorporated in the proposal.

In summary, current security extensions to OLSR cover a sizeable number of distinct problems. Consensus seems to have been reached in the use of signature and key management systems to ensure the integrity and authenticate the sender of routing control traffic. Similarly, timestamps have found full acceptance in the referred proposals dealing with the replay of old messages. For the remaining issues, however, different techniques have been proposed. In the case of link spoofing by compromised nodes, the techniques presented vary from establishing a line of defense between trusted and untrusted nodes, to the transmission of a cryptographic message in conjunction with routing control traffic. For incorrect traffic relaying, proposals are based on detecting misbehavior based upon the number of packets sent and received by each node or by

the usage of geographical positioning.

3.4 Summary

The OLSR protocol is based on a multipoint relay optimization of the classical link state algorithm. This optimization results in a minimization of flooding, because only a selected set of neighbor nodes are used to diffuse information through the network, and in a reduction of the size of control packets, because only a subset of links needs to be declared. The protocol assumes that all nodes are benign in the exchange of topology information and, therefore, is subject to several vulnerabilities. Various proposal to secure OLSR were discussed. Although these proposals solve some of the key security issues, improvements can be made by scrutinizing the underlying assumptions and the aforementioned technical drawbacks. Thus, while adopting some of the generally accepted schemes for tasks such as avoiding replay attacks or guaranteeing integrity and authentication, we will propose a reputation-based security scheme to address the generation of fake routing control traffic.

Chapter 4

A Reputation-based Security Scheme for OLSR

In this chapter we provide the motivation for our work, followed by a description of the type of attacker we consider. Subsequently, we thoroughly describe our reputation-based security scheme to tackle the defined attacker and address the identified remaining security issues.

4.1 Problem Statement

From the taxonomy of security vulnerabilities and the previous work on securing the OLSR protocol, we have identified two types of vulnerabilities for which there are commonly accepted solutions: identity spoofing attacks can be tackled with signature and key management systems, and replay attacks can be addressed with a timestamp mechanism.

The motivation for this work is the need to find a scheme to address the remaining security issues, which (a) does not depend on computationally heavy cryptographic primitives, (b) does not impose a large overhead to the network in terms of additional traffic and, naturally, (c) properly punishes or restricts misbehaving nodes actions.

4.1.1 Attacker Model

For know we will focus on the prevention of the generation of fake routing control traffic, namely the link spoofing attack.

Thus, we consider an active attacker. This attacker injects packets into the network with the intent of disrupting or adjusting the routing protocol operation according to his will. We assume that the attacker is not able to either impersonate other nodes (this can be prevented through the use a distributed certificate authority such as those presented in [ARM05, DRWL04, Raf05]) nor to replay old messages in the network (to prevent this, a timestamp mechanism such as the ones in [ACJ+03, ACL+05] can be employed).

The behavior of the attackers defined and the corresponding consequences to the network is the following.

HELLO link faker

This attacker performs link spoofing by adding the fake information that he is able to reach all of his two-hop neighbors with the intent of forcing the selection as a MPR. This attack may be harmful in two ways [Raf05]: (a) it can cause the selection of a wrong MPR set and (b) messages sent by the attacked node may not reach some of his two-hop neighbors.

TC link faker

This attacker performs link spoofing by randomly choosing one or more distant nodes in the network and announcing direct connectivity to them. This attack may be harmful because it introduces conflicting routes and promotes loss of connectivity and increase in path lengths in the network.

4.2 Security Scheme

In this section we will describe our Cooperative Security Scheme for OLSR (a very basic version of it was presented in [VB06]), along with the challenges that arose during the implementation and validation process and their impact on the security scheme

operation.

The fundamental concern behind the Cooperative Security Scheme for OLSR (CSS-OLSR) is that of assuring that nodes correctly generate OLSR control traffic. To achieve this goal, the guiding principle of CSS-OLSR is to reward nodes that comply with the routing protocol and penalize damaging behavior [BB02, MM02a] in terms of network availability, i.e. by reducing the ability for malicious nodes to communicate through the network.

For this purpose, we add two new elements to regular OLSR operation:

- *Complete path message (CPM):* A CPM is used to convey the path traversed by another message through the network. Upon receipt of a TC message, according to the rules specified below, each MPR node sends a CPM back to the originator of the TC with the path traversed by the TC message which, therefore, records the path traversed by itself on its payload;

- *Rating table:* Each node of the network keeps a rating table which holds information about the behavior of nodes in the network. Each entry in the rating table has a node ID, a primary and secondary ratings. The node ID uniquely identifies a node in the network, the secondary rating is a classification of the node based on the direct observation of packet retransmissions, and the primary rating is a more mature classification of the node based the correlation of its secondary rating, the analysis of information provided by CPMs and local routing information kept by the nodes.

Since our security scheme relies upon the ability to uniquely identify each node and the exact origin of each packet, we assume the use of a distributed certificate authority (*DCA*) that conforms with the MANET paradigm such as those presented in [ARM05, DRWL04].

In order to ensure the integrity of the path stored in the TC messages and subsequently in the CPM messages, a cryptographic mechanisms such as those presented in [HPJ05] to protect against the tampering of routes of on-demand routing can be relied upon. Additionally, a timestamp mechanism such as those previously mentioned protects against the repeat of old messages.

4.2.1 Protocol Specification

A security extension to the OLSR protocol that employs the proposed scheme can be defined as follows.

1. At the formation of the network, a *DCA* is employed guarantying the proper authentication of each node;
2. During the broadcast of HELLO messages to ensure knowledge of one and two-hop neighbors, only properly authenticated nodes are considered;
3. For each authenticated node found, a new entry in the rating table is added with value α for the secondary rating and ρ for the primary rating;
4. Using the information from HELLOs, each node performs the selection of their MPR set, which is announced in subsequent HELLO messages;
5. Using this information, each node constructs its MPR selector set with the nodes that selected it as a MPR;
6. A TC is periodically flooded in the network by each node, declaring its MPR selector set;
7. A mechanism based on the *watchdog* concept [MGLB00] is employed to detect misbehavior through direct observation of TC retransmissions;
8. Upon receipt of a TC message, a CPM containing the path traversed by the TC message may be sent back to the origin depending on the rate λ of CPM transmission;
9. Using the information of the TCs received, each node maintains a topology table which consists of entries with an address of a destination (a MPR selector in the TC message), an address of a last-hop node to that destination (the originator of the TC) and a MPR selector set sequence number;
10. When a CPM is received, it is processed according to the Algorithm 1 for CPM processing;
11. The topology table is then used by the routing table calculation algorithm to compute the routing table at each node. Details about this procedure may be found in [CJ03].

Note that steps 4–6, 9 and 11 belong to the regular OLSR operation while the remaining ones are introduced as part of our security scheme.

The initial primary (ρ) and secondary (α) ratings of the nodes on step 3 basically state the level of trust on the network nodes. If we consider a benign network, we can set them to high values, otherwise, by setting them to lower values we are forcing the

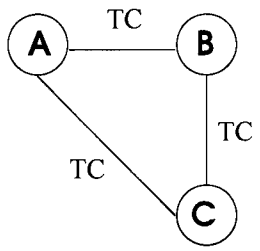

Figure 4.1: Duplicate retransmission avoidance issue

nodes to recover from a misbehaving state from the formation of the network. The optimization of these rating factors will be discussed in Section 5.3.1.

The CPM rate λ on step 8 specifies the amount of CPMs that is generated in response to the reception of TC messages by nodes in the network.

The two following mechanisms for the detection of misbehaving nodes are used.

Detection of Misbehavior Through Direct Observation

The detection of misbehavior through direct observation is based on the *watchdog* concept [MGLB00]. Each node promiscuously listens to its MPR transmissions. If a node detects that a MPR does not relay its packet, it decreases the MPR secondary rating by τ. Otherwise, its secondary is increased by γ.

To encourage cooperation, the punishment should be greater than the reward, i.e. $\tau > \gamma$.

There is a set of issues that bear upon the direct observation of retransmissions [MGLB00] which make it error-prone, such as packet collision, limited transmission power, node collusion and partial packet dropping. Apart from these well-known issues, there is one which is specific to the OLSR operation. We call this issue the *duplicate transmissions avoidance issue* because it derives from the duplicate transmission avoidance mechanism mentioned in Section 3.1, where duplicate transmission of control traffic is avoided by keeping a record of received messages in a *duplicate set*. This mechanism affects the detection of the misbehavior through direct observation in the following way.

Regard the scenario in Figure 4.1. Consider that B and C are MPRs of A, and C is MPR of B. At time instant 0, A sends a TC message which is received by nodes B and C. At time instant 1, B and C forward this message because they are MPRs of A. When B does so, it expects C to subsequently forward it because C is also MPR

of his. Although, due to the duplicate transmission avoidance mechanism B does not forward the message from C because he already did it when he received the message sent from A at time instant 0. This results in a false detection of misbehavior because B was actually just following the regular OLSR operation.

The approach to solve this issue is the following. The *duplicate set* holds unique identifiers of messages received which are used by the duplicate transmission avoidance mechanism. Without changing the duplicate transmission avoidance mechanism we just keep the additional information of which nodes sent those messages. This information allows us to subsequently determine whether a node has already sent a message we are forwarding and, therefore, prevent erroneously accusing nodes that are just following the protocol rules, as in the example above.

Detection of Misbehavior Through Analysis of the CPMs

Although OLSR assumes a bidirectional connection between a node and its MPRs, as stated before, the detection of misbehavior through direct observation is error-prone. Moreover, the detection of misbehavior through direct observation would only allow each node to punish neighboring nodes and we aim at punishing misbehaving nodes on a network-wide perspective, without using reputation dissemination, (because it allows to falsely accuse well-behaved nodes).

Thus, the secondary rating (obtained through direct observation of a neighbor node's retransmissions) is solely an unreliable node status and is only used to dictate how fast a node can recover from a misbehavior state, as considered by other nodes.

Algorithm 1 CPM processing

1: $SR_s \leftarrow$secondary rating of the node under scrutiny, S
2: $PR_s \leftarrow$primary rating of the node under scrutiny, S
3: **if** the path information in the CPM is not coherent with the local information obtained from control messages from S **then**
4: $\quad PR_S \leftarrow PV$
5: **else**
6: $\quad$ **if** $SR_S < PR_S$ **then**
7: $\quad\quad SR_S \leftarrow SR_S + SRV$
8: $\quad$ **else**
9: $\quad\quad PR_S \leftarrow PR_s + PRV$
10: $\quad$ **end if**
11: **end if**

To classify nodes as misbehaving the primary rating is used. A default primary rating is attributed when a new node is first detected and the variations on this rating are a result of two mechanisms: (a) the detection of fake control traffic generation mechanism (step 3 of the CPM processing algorithm), and (b) the recovery from misbehavior mechanism (steps 6–9 of the CPM processing algorithm).

Due to the different nature of the two types of control traffic messages of OLSR that can be faked, the nodes under scrutiny in the CPM processing algorithm are, in the detection of fake HELLO messages the MPRs of the current node, and in the detection of fake TC messages all the nodes in the path contained on the CPM. The reason for this will become more clear when in the following sections we describe the mechanisms to detect (a) fake HELLO generation and (b) fake TC generation (step 3 of the CPM processing algorithm), (c) to punish malicious nodes (step 4) and (d) to recover from a misbehaving state (steps 6–9).

4.2.2 Detection of Fake HELLO Generation

For the detection of fake HELLO generation, two sources of information are used: the paths obtained from CPM messages, and the local information obtained from HELLOs kept in the *neighbor 2-hop set*.

Let us consider the scenario of Figure 4.2 in which node C generated a TC message and is now receiving a CPM from one node in the network. Let M be a MPR of C which lies in the path of the CPM (i.e. M was the forwarder of the TC from C which originated the current CPM). Moreover, let T be a node at two or more hops from M in the path of the CPM. The procedure to detect fake HELLO generation is the following.

1. C receives a CPM which holds the path of a TC message sent by him to the

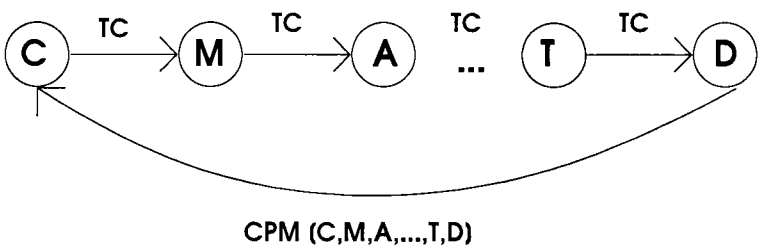

Figure 4.2: CPM scenario

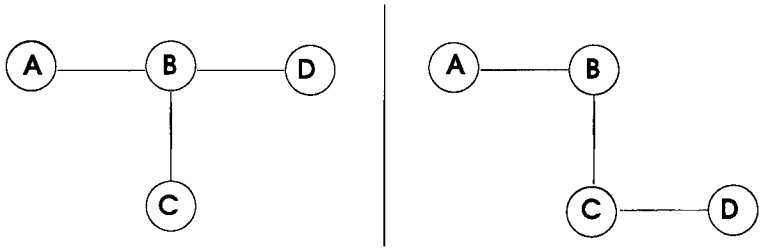

Figure 4.3: MPR transient state issue

network;

2. C checks, for every node T two or more hops away from M, if there is an entry in the neighbor 2-hop set stating that M has direct connectivity to T;

3. If so, then M is a misbehaving node because he announced direct connectivity to T through HELLO messages and T is not directly reachable by M;

4. Otherwise M is considered a well-behaving node;

5. Taking in consideration if M is a misbehaving node or not, appropriate measures are taken by the punishment mechanism (see Section 4.2.4) and the recovery mechanism (see Section 4.2.5).

There is one issue with this approach. The local information kept by OLSR nodes is based on a periodic exchange of control traffic. With nodes moving, there are transient states where the actual network state and the local information are not coherent.

Regard, for example, the scenario of Fig. 4.3. Consider that on the left side, node B is a MPR of A, and C and D are MPRs of B. D is now moving and gets out of the transmission range of B and into the transmission range of C, becoming a MPR for C (right side of Fig. 4.3). In the meanwhile, the periodic exchange of control traffic did not occur and, therefore, A is still not aware of this topological change. A sends a TC to the network which follows the path A-B-C-D and D generates a CPM containing this path. Since the local information of A states that B can reach D (because the local information of A was not updated yet), this results in a false positive of misbehavior detection where B is the misbehaving node.

One possible solution for transient states is strengthening the requirement for accepting a node as a symmetric neighbor [CHCB01], e.g. by requiring that a number of HELLO messages is received within a certain time-frame for a link to be considered as valid. The cost of this solution would be a slower response to the appearance of new links.

From the simulation results we were able to see that these false positives are sparse and much less frequent than the correct detections of misbehavior. Taking this into consideration, we developed a recovery mechanism (see also Section 4.2.5) to solve this issue by recovering nodes from the false detection of misbehavior and still properly punishing misbehaving nodes.

4.2.3 Detection of Fake TC Generation

The detection of fake TC generation is based on two sources of information: the paths obtained from CPM messages and the local information from TC messages kept in the *topology set*.

Let us consider the node C as a network node which is receiving a CPM from the network. The procedure to detect fake TC generation is the following.

1. C receives a CPM which holds the path of a TC message sent to the network by some node;
2. For every node M in the CPM path and every node T three or more hops away from M also in the path, C checks if there is an entry in the *topology set* stating that M has direct connectivity to T;
3. If so, then M is a misbehaving node because he announced direct connectivity to T through TC messages and T is not directly reachable by M;
4. Otherwise M is considered a well-behaving node;
5. Taking into consideration if M is a misbehaving node or not, appropriate measures are taken by the punishment mechanism (Section 4.2.4) and the recovery mechanism (Section 4.2.5).

The detection of fake TC generation is also affected by the MPR transient state problem mentioned in the previous section. In this case, we cannot rely on the recovery mechanism because it is based on direct interaction between nodes (see Section 4.2.5) which can delay the recovery of well-behaving nodes and this is not acceptable.

Our approach to tackle this problem was already described in step 2 of the procedure to detect fake TC generation above. Basically, instead of analyzing the connectivity of nodes which are two or more hops away from the eventual malicious node, we analyze it

for nodes three or more hops away, which greatly reduces the number of false positives by reducing the number of occurrences of the MPR transient state. This approach allows a malicious node to fake connections to nodes at two hops away, although we find this a reasonable compromise because of the very low number of false positives obtained and, by faking connections to nodes at two hops away, a malicious node is only able to increase the path length by 1.

4.2.4 Punishment of Malicious Nodes

After detecting if a node is misbehaving, proper measures must be taken. Basically, as seen in step 4 of the CPM processing algorithm, the primary rating of the malicious node is set to a Punishment Value (PV).

The primary rating ranges from 0 to 100. In order to motivate nodes to behave well, the primary rating is then used by the network nodes to determine their willingness to forward traffic for other nodes. This is done by relaying other nodes' traffic according to their primary rating. E.g. a node A has a primary rating of 50 for a node B, therefore A will randomly discard half of the packets from B, this way forcing node B to behave correctly in order for it to be able to effectively communicate through A.

4.2.5 Recovery From Misbehavior State

The recovery mechanism allows a node that stops misbehaving to recover from the misbehavior state. This mechanism is also used to address the false detections of misbehavior mentioned in Section 4.2.2.

We call this mechanism *direct interaction recovery* since it is only active when nodes interact directly, i.e. only when a node is near another he is able to recover from a misbehavior state. The reason for this choice is twofold. Firstly, we need a mechanism that allows well-behaved nodes to recover from the false detections of fake HELLO and, therefore, this mechanism has to allow nodes to recover through direct interaction because the detection of fake HELLO is only done among neighboring nodes. Secondly, from our simulation results we were able to see that the amount of CPMs leading to a detection of good behavior is much larger that the amount of CPMs leading to detection of misbehavior and, therefore, we needed to restrict the amount of CPMs that foster the recovery of nodes, otherwise malicious nodes would recover too fast and would not be properly punished.

Our *direct interaction recovery* mechanism entails a slow recovery of nodes which are found to refuse relaying control traffic on behalf of other nodes. The overall procedure, as seen in steps 6–9 of the CPM processing algorithm is the following. If the secondary rating of the recovering node is lower than its primary rating, only the secondary rating is increased by SRV (Secondary Recovery Value) until it reaches the value of the primary rating. This buffer of time delays the recovery of nodes which have been refusing to relay control traffic because only once the secondary rating reaches a value larger than the primary rating the misbehaving node effectively starts recovering by having an increase of PRV (Primary Recovery Value) to the primary rating.

While this approach based on direct interaction may not be well suited for every kind of networks (e.g. if two nodes do not move and accuse each other, they will never be able to recover if they are not within range of one another), we claim that this is not a problem since other type of mechanisms can be used on top of this which are not based on the proximity of the involved nodes, e.g. a timeout mechanism can be used to allow nodes to recover after a reasonable amount of time.

4.3 Summary

Based on the taxonomy of vulnerabilities of OLSR and the existing security solutions, in this chapter we provided the motivation for our work, along with the attacker model we consider. Then, we described our security scheme used to counter the defined attacker. We also described relevant issues that arose during the implementation and validation process, which led to some modifications of the original specification of CSS-OLSR.

Chapter 5

Performance Evaluation

In this chapter we present an analysis of the overhead and evaluate the effectiveness of CSS-OLSR in the presence of malicious nodes in the network. The evaluation of the overhead is made both analytically and through simulations. The impact in terms of the reputation of the nodes in the network was determined under different simulation scenarios.

5.1 Problem Statement

Our security scheme for OLSR introduces two new components to the regular OLSR operation: the rating tables and the CPMs. The first component does not introduce any communication overhead in the network, only some minimal processing overhead as result of the mechanisms which lead to updates on the ratings of the nodes. As of the CPMs, a thorough analysis of the overhead in terms of network transmissions is needed in order to determine the impact and applicability of the proposed security scheme.

This will be done in two complementary ways: through an analytical evaluation based on random graph theory, and through a simulation-based evaluation based on protocol implementation.

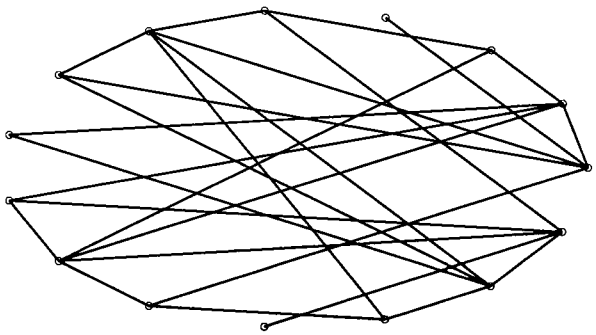

Figure 5.1: Random Graph – 15 nodes, 0.8 link probability

5.2 Analytical Evaluation

In this section we present a mathematical comparison between the overhead of the classical OLSR protocol and the one that results from the modifications related to our Cooperative Security Scheme.

In [JLMV01], the performance of the classical OLSR is evaluated. Focusing on the multipoint relay concept, a comparison with standard link state protocols is made. The evaluation is done in two radio network models from graph theory: the random graph and the random unit graph (also called random geometric graph). The first is more suitable for representing indoor networks, whereas the second is more suitable for outdoor networks.

5.2.1 Network/Graph Models

Our evaluation goes along the lines of the one performed in [JLMV01] and, therefore, the same graph models are used.

Random Graph Model

The most commonly studied random graph model is the $G(N, p)$, where N is the number of vertices and p is the link probability. This model basically states that a link exists between any two nodes with link probability p and they are independent among themselves. See, for example, Figure 5.1 for a random graph with 15 nodes and a link probability of 0.8.

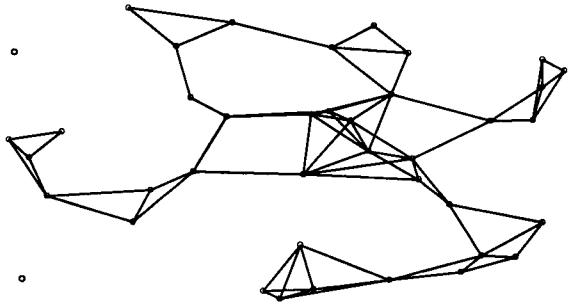

Figure 5.2: Random Unit Graph – 40 nodes, 0.2 radius

Random Unit Graph Model

The random unit graph is characterized by the number of vertices N, the unit-length L and the dimension. This graph is obtained by randomly placing the N nodes in an area and systematically linking all the pairs of nodes which are at a distance smaller of equal to the unit-length L. The area where the nodes are placed varies with the dimensions and unit-lengths considered. See, for example, Figure 5.2 for a 2-dimensional graph with 40 nodes and a connection radius of 0.2.

5.2.2 Terminology and Previous Work

Both the random graph and the random unit graph models are instances of a general graph concept defined by a set of vertices (or nodes) V, and a set of connections between those vertices $E \subseteq V \times V$.

In computer networks, the vertices are usually called "network nodes" (or just nodes), and the connections between them are referred to as "links". Since OLSR only considers bi-directional links in its communication, a node in the graph is connected to another if they are both on the transmission range of one another, i.e. if they can hear each other.

In a link state routing protocol such as OLSR, each node performs two tasks: neighbor discovery and topology broadcast. In a plain link state protocol, neighbor discovery means that each node periodically broadcasts hello packets announcing its one-hop neighbors to adjacent neighbors; topology broadcast is performed simply by having each node periodically broadcast the list of all its neighbors in a topology packet and

every other node in the network re-broadcasting this packet (full flooding).

Let us consider h the rate of hello packets transmission and τ the rate of topology packets transmission. Let N be the number of nodes in the network and M be the average number of adjacent links per node.

The overhead in terms of packets transmitted Ω_{pkt} by a routing protocol is given by

$$\Omega_{pkt} = hN + \tau R_N N,$$

which in terms of IP addresses transmitted becomes

$$\Omega_{ip} = hNM + \tau N R_N D_N,$$

where D_N is the average number of links to be announced in topology packets per node, and R_N is the average number of retransmissions in the flooding of topology packets.

For a plain flooding link state protocol (e.g. OSPF) the overhead is [JLMV01], in terms of packets transmitted, $hN + \tau N^2$, and in terms of IP addresses transmitted $hNM + \tau N^2 M$, i.e. $R_N = N$ — the average number of retransmissions in the flooding of topology packets is the total number of nodes of the network, since every topology packet transmitted by each node is retransmitted by every other node in the network (full flooding); and $D_N = M$ — the average number of links to be announced in topology packets is the average number of adjacent links per node, since all adjacent links should be announced.

In OLSR, the average number of links to be announced in topology packets D_N is the average number of MPR links per node, since only MPR links are announced, and the average number of retransmissions in the flooding of topology packets R_N is the average number of retransmissions made by MPR nodes, since only these nodes are allowed to retransmit topology packets.

Bounds were calculated for both D_N and R_N for OLSR that favorably compares to plain link state routing protocols in terms of control traffic overhead. Refer to [JLMV01] for details about the analysis made.

From now on, we will present the results for the OLSR control traffic overhead in the aforementioned network models and compare them with our results in the analysis of the CSS-OLSR.

5.2.3 Analysis in the Random Graph Model

In the random graph model, the expressions for the control traffic overhead are $hN + \tau N R_N$ in terms of packets transmitted, and $hNM + \tau N R_N D_N$ in terms of IP addresses transmitted.

From [JLMV01] we know that the average number of MPR links per node D_N, and the average number of retransmissions in an MPR flooding R_N in the random graph model are both $O(\log N)$.

Therefore, the cost of the classical OLSR control traffic is $O(N \log N)$ in terms of packets transmitted, and $O(N^2)$ in terms of IP addresses transmitted. Notice that in the cost in terms packets transmitted the dominant source of traffic overhead is the topology broadcast mechanism, whereas in the cost in terms of IP addresses transmitted the dominant source of overhead is the neighbor sensing mechanism.

The following Lemma will be used later on to determine the length of the longest optimal path of the network in the random graph model.

Lemma 1. *[JLMV01] With fixed edge probability p, the optimal route between two random nodes in a random graph, when N tends to infinity,*

(i) is of length 1 with probability p;

(ii) or of length 2 with probability $q = 1 - p$.

Theorem 1. *The cost of the CSS-OLSR control traffic for CPM mechanism in the random graph model is $O(N \log N)$.*

Proof. With the inclusion of the Complete Path Messages (CPMs) in the protocol, a new set of parameters needs to be added to the control traffic overhead expression in order to consider the additional impact of these messages in the protocol. The expression for the control traffic overhead in terms of packets transmitted Ω_{pkt} considering the CPM mechanism is then

$$\Omega_{pkt} = hN + \tau N R_N + \rho(\tau N R_N)\Delta_N,$$

where the first two elements are the same as for the classical OLSR, ρ is the rate of CPM generation, and Δ_N is the length of the longest optimal path in the network. The third element represents the overhead of the CPMs, which is given by the rate of CPM generation $\rho(\tau N R_N)$, times the length of the longest optimal path in the network Δ_N.

In terms of IP addresses transmitted, the expression for the control traffic overhead is

$$\Omega_{ip} = hNM + \tau N R_N (D_N + \Delta_N) + \rho(\tau N R_N)(\Delta_N)^2,$$

where the first element is the same as for the classical OLSR and the others consider the number of IP addresses stored due to keeping the traversed path information in both the topology packets and the CPM messages. For that, in the second element we add the length of the longest optimal path Δ_N (which is the top bound for the amount of IP addresses kept due to storing the paths traversed by the messages) to the average number of links to be announced in topology packets D_N. As of the CPM overhead, the maximum number of IP addresses carried due to storing the paths traversed by the messages is also Δ_N, since the paths stored in the CPMs are copied from the topology packets.

From Lemma 1, we know that, as the number of nodes tends to infinity, the length of the longest optimal path between two random nodes in a random graph is expected to become constant and, therefore, has no impact on the order of the average number of retransmissions of CPMs, which is then $O(N \log N)$, both in the packets transmitted and in the IP addresses transmitted cases, and this ends the proof.

□

Notice that the cost of the CPM mechanism in the random graph model:

(i) is of the same order of the cost of the overhead of the classical OLSR control traffic in terms of packets transmitted, with cost $O(N \log N)$;

(ii) is lower than the cost of the overhead of the classical OLSR control traffic in terms of IP addresses transmitted, with cost $O(N^2)$.

5.2.4 Analysis in the Random Unit Graph Model

In the random unit graph model, the expressions for the control traffic overhead are the same as in the random graph model. We just have to recalculate the bounds for the average number of MPR links per node D_N, for the average number of retransmissions in an MPR flooding R_N, and for the length of the longest optimal path in the network Δ_N.

The expressions for the control traffic overhead are then, in terms of packets transmitted $hN + \tau N R_N + \rho(\tau N R_N)\Delta_N$, and in terms of IP addresses transmitted $hNM +$

$\tau N R_N(D_N + \Delta_N) + \rho(\tau N R_N)(\Delta_N)^2$.

One-Dimensional Random Unit Graph

A 1D random unit graph is obtained by linking pairs of nodes whose distance is smaller or equal to the unit length (typically the transmission range of the nodes) when N nodes are uniformly distributed on a strip with L units length.

From [JLMV01] we know that for the classical OLSR the number of MPR links per node D_N is either 1 or 2, and that the MPR flooding of a message takes $R_N = \lfloor L \rfloor$ retransmissions when N tends to infinity and L is fixed.

Therefore, the cost of the classical OLSR control traffic is $O(NL)$ in terms of packets transmitted, and $O(N^2/L)$ in terms of IP addresses transmitted.

The following Lemma will be useful to determine the cost of the CPM mechanism in the random unit graph model.

Lemma 2. *[JLMV01] The MPR flooding of a broadcast message originated by a random node takes $\lfloor L \rfloor$ retransmissions of the message when the number of nodes N tends to infinity.*

Theorem 2. *The cost of the CSS-OLSR control traffic for the CPM mechanism in the 1D random unit graph model is $O(NL^2)$ in terms of packets transmitted, and $O(NL^3)$ in terms of IP addresses transmitted.*

Proof. In terms of packets transmitted, the cost of the CSS-OLSR control traffic for the neighbor sensing mechanism and the topology broadcast mechanism is the same as in the classical OLSR, respectively $O(N)$ and $O(NL)$.

From Lemma 2 we know that the topology broadcast of a message takes $R_N = \lfloor L \rfloor$ retransmissions when the number of nodes N tends to infinity. These number of retransmissions corresponds to a message traversing all MPR nodes of the network through the optimal path and, therefore, the length of the longest optimal path in the network is at most $\Delta_N = \lfloor L \rfloor$.

The cost in terms of packets transmitted of the CPM mechanism in CSS-OLSR is then $O(NL^2)$.

In terms of IP addresses transmitted, the cost of the CSS-OLSR control traffic for the neighbor sensing mechanism is $O(N^2/L)$ – the same as for the classical OLSR, the

cost for the topology broadcast mechanism is $O(NL)$, and for the CPM mechanism the cost is at most $O(NL^3)$, and this ends the proof.

□

Notice that the cost of the CPM mechanism in the one-dimensional random unit graph:

(i) adds the constant factor L to the overhead of the classical OLSR control traffic in terms of packets transmitted $O(NL)$, and constant terms are negligible when considering the order;

(ii) is lower than the cost of the overhead of the classical OLSR control traffic in terms of IP addresses transmitted, with cost $O(N^2/L)$.

Two-Dimensional Random Unit Graph

The 2D random graph is similar to the previous but, instead of having nodes distributed on a strip of length L, they are distributed in a square of dimensions $L \times L$.

From [JLMV01] we know that for the classical OLSR when L is fixed and N increases D_N tends to be smaller than $3\pi(N/(3L^2))^{1/3}$, and that R_N is $O((NL^4)^{1/3})$.

Therefore, the cost of the classical OLSR in terms of packets transmitted is $O((NL)^{4/3})$ from the topology broadcast mechanism, and in terms of IP addresses transmitted $O((N/L)^2)$ from the neighbor discovery mechanism.

Lemma 3. *The length of the longest optimal path for the 2D random unit graph is $\sqrt{2}L$ hops when the number of nodes N tends to infinity.*

Proof. In a $L \times L$ random unit graph the distance between a node and its MPRs tends to be equal to one unit length when the density increases. Therefore, the length of the longest optimal path is given by the distance in terms of unit lengths between the two most distant points of the square.

These points are the ones located at two opposite edge vertices of that square, which are at a distance of $\sqrt{2}L$, and this ends the proof. □

Theorem 3. *The cost of the CSS-OLSR control traffic for the CPM mechanism in the 2D random unit graph model is $O(N^{4/3}L^{7/3})$ in terms of packets transmitted, and $O(N^{4/3}L^{10/3})$ in terms of IP addresses transmitted.*

Proof. In terms of packets transmitted, the cost of the CSS-OLSR control traffic for both the neighbor sensing mechanism and the topology broadcast mechanism is the same as in the classical OLSR, respectively, $O(N)$ and $O((NL)^{4/3})$.

From Lemma 3 we know that the length of longest optimal path is $O(L)$.

The cost in terms of packets transmitted of the CPM mechanism in CSS-OLSR is then $O(N^{4/3}L^{7/3})$.

In terms of IP addresses transmitted, the cost of the CSS-OLSR control traffic for the neighbor sensing mechanism is the same as in the classical OLSR – $O((N/L)^2)$, the cost for the topology broadcast is $O(N^{5/3}L^{2/3})$, and the cost of the CPM mechanism is $O(N^{4/3}L^{10/3})$, which ends the proof. □

We observe that the cost of the CPM mechanism in the two-dimensional random unit graph has the following properties:

(i) it adds the constant factor L to the overhead of the classical OLSR control traffic $O((NL)^{4/3})$, and constant terms are negligible when considering the order;

(ii) it is lower than the cost of the overhead of the classical OLSR control traffic in terms of IP addresses transmitted, with cost $O((N/L)^2)$.

5.3 Simulation-based Evaluation

In this section we present a simulation-based evaluation evaluation of CSS-OLSR and a comparison in terms of traffic overhead with the classical OLSR protocol.

The drawbacks of simulation studies, namely for MANETs [KCC05], are widely known. Therefore, our goal with these results is merely to support (not prove!) the proposed concepts and provide a demonstration of the operation of our security scheme.

5.3.1 Design of Experiment

The simulations where performed using the ns2 network simulator [Sim05] version 2.29.2 with a modified version of UM-OLSR [RR04] implementation version 0.8.8 of OLSR (the code for the modified version is available in [Vil06]). All the default values for the OLSR protocol from RFC3626 [CJ03] of OLSR where used.

Type of Simulation

The simulations performed are of the steady-state type, i.e. we are interested in long-run average behavior of ad hoc networks (independent of initial simulation conditions) instead of the analysis of the response to a certain network configuration that represents a particular case of operation.

Transient Phase Elimination

During the start-up time of OLSR, nodes do not have any information about the network, their tables are empty and they start sending HELLO and TC messages to disseminate connectivity information through the network. From [Qay00] we know that, independently from the number of nodes in the network, OLSR takes from 5 to 6 seconds to get stabilized in terms of route availability. This should only be a concern when analyzing the graphs and not when performing the statistical analysis because our statistical analysis is done for each time instant, thus not being affected by previous results.

Number of Runs and Confidence Intervals

Since routing protocols are very sensitive to movement patterns and network topologies, we generated scenarios with 10 different movement patterns.

To average the results and diminish the choice of a favorable or unfavorable pick of scenarios, the method of *independent replications* shown in [GT00] was used. Five independent replications were run, each with a set of 10 distinct mobility scenarios, which results in a total of 50 simulation runs for each set of parameters under evaluation.

Afterwards, we calculated the average of the measures measures of interest and got the corresponding 95% confidence intervals.

Random Number Generator (RNG)

In the ns2 network simulator, when a new RNG object is created, it is automatically seeded to the beginning of the next independent stream of random numbers, which allows for many different random variables [FV06]. In order to run several independent

replications of the simulations and subsequently perform statistical analysis given multiple runs without correlation of a set of scenarios, we have set the default RNG seed to a different number in each set of mobility scenarios, which results in any other RNGs being automatically seeded such that they produce independent streams.

Mobility Model

We used the random waypoint mobility model for our simulations. It goes as follows. At each trip transition instant a node uniformly picks a destination at random from rectangular area. It then picks a movement speed from a uniform distribution and travels from the current position to the new one at the specified speed. When reaching the destination the node pauses for a random time also retrieved from a uniform distribution.

To address the initialization bias associated with initial node movement we used the tool from [PBV05], which produces a perfect sampling of the node mobility state (which is then used as input to ns2), so that the simulation of the node mobility is stable (i.e. in a steady state) throughout all the simulation.

Mobility Speeds and Pause Intervals

In order to exercise a network with mobile nodes, we considered the node speeds of 1.4m/s and 2.4m/s. The impact of two different pause intervals (1 and 5 seconds) for the mobility model explained above was also evaluated.

In OLSR, the routing table is updated in case of neighbor appearance or loss. Therefore, the mobility speed is a particularly important parameter since it causes a significant change in the protocol performance by means of changing the connectivity in the network. We have chosen these two values since they exercise a network with mobile nodes (respectively, 5 kilometers per hour and 8.6 kilometers per hour). Values above the ones used result in exaggerated topology/connectivity changes and require alternative solutions (e.g. [BMA02]) in order to keep the network topology information up-to-date.

Scenarios

The simulations were performed for 30 nodes with a transmission range of 250 meters, in a area of size 1500x300 meters during 900 seconds. The choice for this scenario was based on the following. Throughout all the iteration runs of the simulations, the average number of hops of the shortest paths was within the interval $[2.23, 2.56]$ for the node speed of 1.4m/s and within the interval $[2.12, 2.43]$ for the node speed of 2.4m/s. Although the intervals for the average number of hops shown are very large, we observed that throughout every iteration of the simulations the amount of paths with three or more hops in the routing tables (the shortest paths) were between 35% – 43% of the total paths for the speed of 1.4m/s and between 31% – 40% for the speed of 2.4m/s. This means that a reasonable amount of messages are relayed when diffused to the network, therefore effectively testing our security scheme.

The attacker considered, follows the behavior presented in Section 4.1.1 and it is active from the 50 to the 300 seconds of simulation time. The two types of attackers (*HELLO link faker* and *TC link faker*) were tested separately. The *TC link faker* was tested for two cases: generating messages with one single fake link, and generating messages with four fake links.

CSS-OLSR parameters

Recall that in our security scheme, the secondary rating is only used to dictate how fast a node recovers from a misbehaving state. Since the goal of CSS-OLSR is to properly punish the generation of fake routing control traffic independently of whether a node refuses to relay traffic, the parameters related to the mechanism to handle traffic relay refusal were set to the default values of $SRV = 1$ (secondary rating recovery value), $\gamma = 1$ (secondary rating increase) and $\tau = 2$ (secondary rating increase). This results, as expected, in very high secondary ratings throughout the network because no traffic relay refuser is in place.

As of the remaining parameters, for the initial primary (ρ) and secondary (α) ratings of the nodes, we consider the network to be benign and, therefore, we have set both of them to the top value of 100 points.

For the CPM rate λ it is hard to know in advance which is the best choice, therefore we performed simulations and analyzed the results for several values of CPM rate. The results and conclusions are shown in Section 5.3.2.

The punishment value PV and the primary recovery value PRV must be set together in order to allow a correct punishment of misbehaving nodes but also a reasonable recovery for nodes that start behaving correctly after misbehaving. Our simulations show that the number of false positives in the detection of fake HELLO are more often than in the detection of fake TC, therefore we used a more severe punishment value, $PV = 0$, for the fake TC detection than the one for the fake HELLO detection, $PV = primary\_rating/2$. As of the primary recovery value, setting it to $PRV = 1$ has shown satisfactory results in terms of punishment vs. recovery of the nodes. If set to higher values it will allow a better recovery but a worse punishment, and vice-versa.

5.3.2 Simulation Results

In this section we present and discuss a set of simulation results underlining the effectiveness of our security scheme and the cost in terms of traffic overhead.

Two type of plots are shown: plots with the average primary ratings of the nodes and plots with the overhead induced by the CPM and OLSR operation.

The plots with the average ratings of the nodes show the ratings for all the nodes in the network. The lines on top correspond to the average ratings of all the well-behaved nodes and the lines in the middle correspond to the average rating of the malicious node, for all the CPM rates considered. The average rating R of a certain node A tells us that, if the traffic in the network is evenly distributed, the punishment mechanism will allow, in average, R % of the traffic originated in A to be delivered to the next destination.

The overhead plots basically allow a comparison of the overhead of the CPM mechanism introduced by our security scheme and the overhead of the regular OLSR operation.

Effectiveness of CSS-OLSR

From the plots in Fig. 5.3 and 5.4 we can see that the behavior of the mechanism for detection of fake HELLO does not significantly change either with the different node speeds or with the variations in the CPM rates. What does change is the recovery mechanism, which is faster for higher values of CPM rate for both the node speeds considered. Moreover, the recovery mechanism is also more effective with a lower node speed, which makes sense since it is based on direct interaction and, with the lower

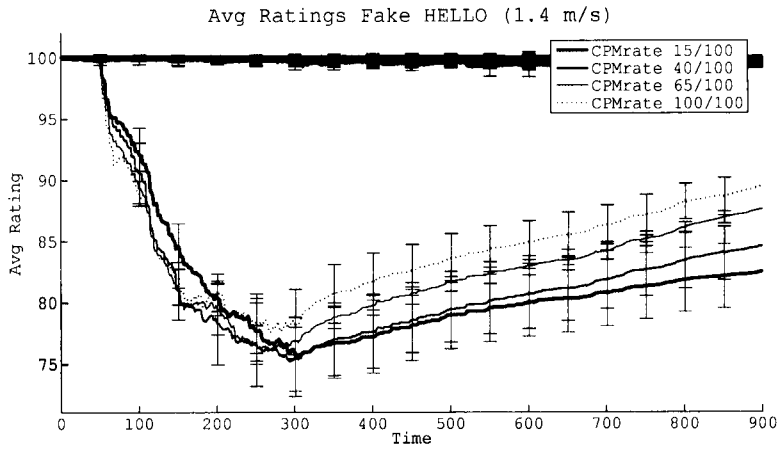

Figure 5.3: Avg. rating of nodes (fake HELLO, 1.4m/s)

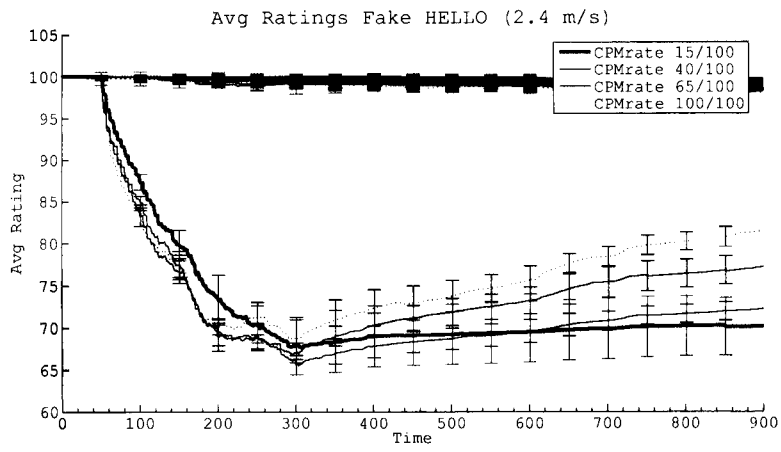

Figure 5.4: Avg. rating of nodes (fake HELLO, 2.4 m/s)

speed, the length of the direct interactions between nodes increases. This fact also justifies why the ratings are slightly lower for a the higher node speed of 2.4m/s.

As of the detection of fake TC, from Figs. 5.5 and 5.6, we can see that this mechanism is already more subject to changes in the CPM rate used. For both node speeds tested, the average rating of the malicious node drops faster and to a lower bottom value for higher CPM rates. The recovery mechanism is also much faster for higher CPM rates. Once again the recovery mechanism works better for the lower node speed of 1.4m/s, in which despite of the fact that the lower primary rating is achieved later that in the 2.4m/s case, it still manages to recover faster and to a higher value at the end of the simulation.

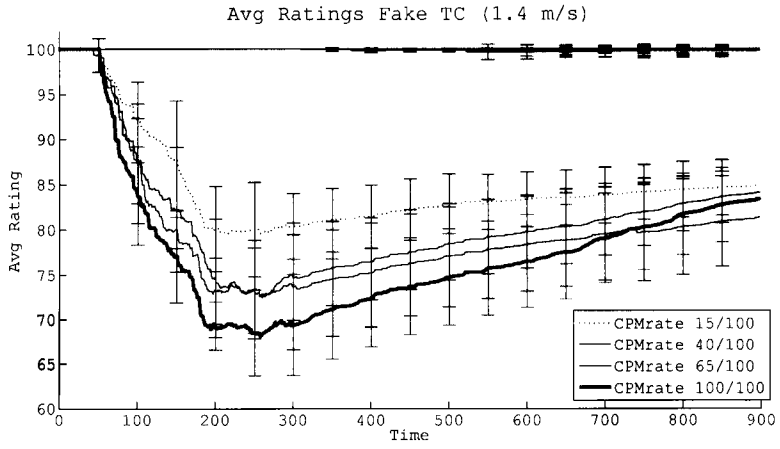

Figure 5.5: Avg. rating of nodes (fake TC, 1.4 m/s)

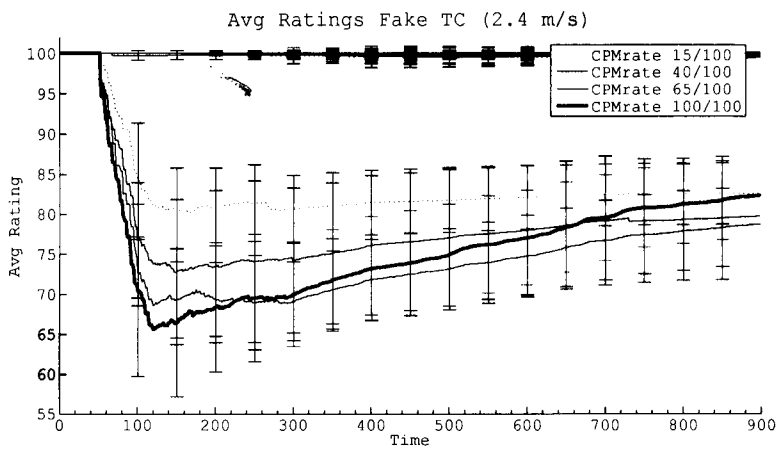

Figure 5.6: Avg. rating of nodes (fake TC , 2.4 m/s)

It may seem that these ratings could be more severe, although it is important to notice that the average ratings presented consider the nodes from the whole network therefore eventually including nodes with which the malicious node does not interact (e.g. because they do not become MPRs and, therefore, do not relay traffic), which results in keeping a high rating for the malicious node. Additionally, for the detection of fake TC, the plots shown consider an attacker that announces a single fake link. As the number of fake links increases, the average primary ratings drop even further. See for example Fig. 5.7 where with 4 faked links the primary ratings drop to lower values than in the previous plots, reaching a value around 55 point for a CPM rate of 100%.

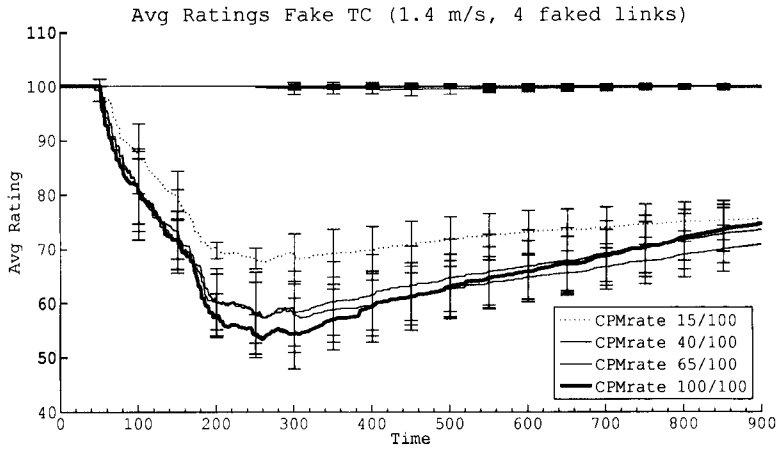

Figure 5.7: Avg. rating of nodes (fake TC, 1.4 m/s, 4 faked links)

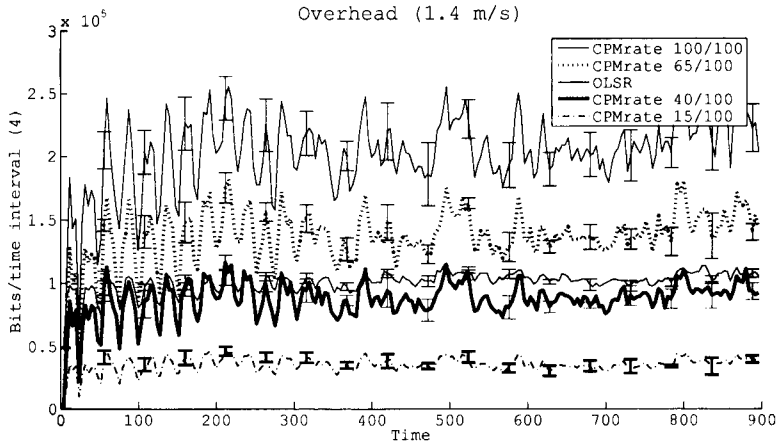

Figure 5.8: Overhead of CSS-OLSR vs OLSR (1.4 m/s)

Overhead of CSS-OLSR

In terms of the overhead results presented in Fig. 5.8, as expected there is a high overhead of our security scheme if a CPM rate of 100% is used and, naturally, as the CPM rate gets lower so does the overhead, becoming very reduced in the case of a CPM rate of 15%. The results for the node speed of 2.4 m/s in Fig. 5.9 are similar to these ones.

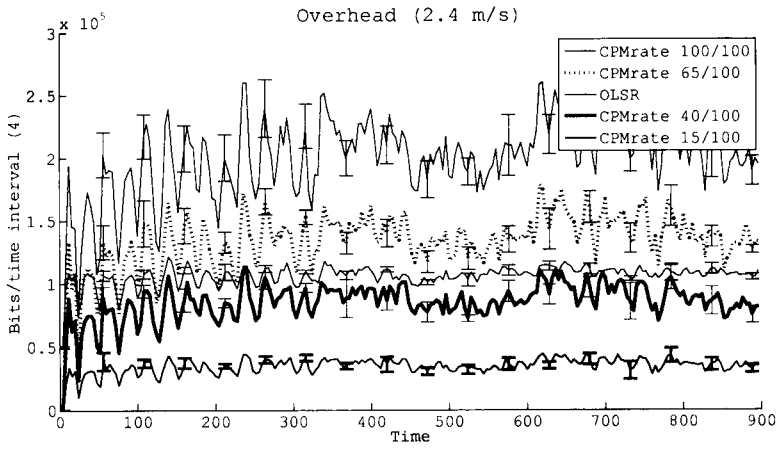

Figure 5.9: Overhead of CSS-OLSR vs OLSR (2.4 m/s)

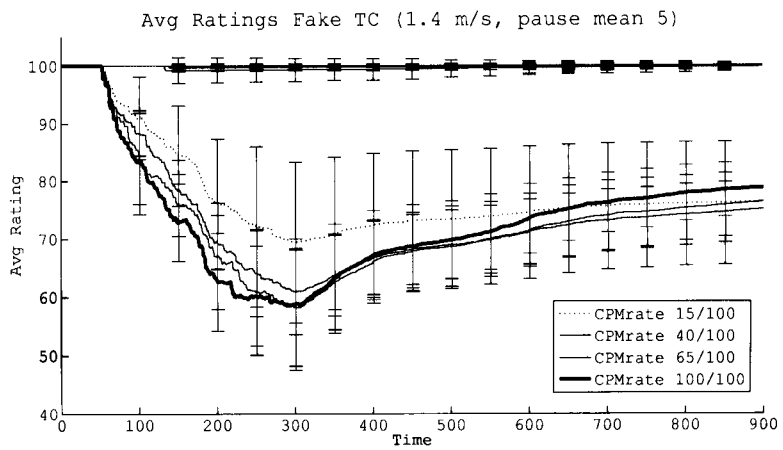

Figure 5.10: Avg. rating of nodes (fake TC, 1.4 m/s, 1 faked link, pause mean 5)

Other Considerations

From the results on Figure 5.10, where a pause mean of 5 seconds was used (instead of the 1 second used in the previous plots), we can see that (1) the malicious node is more severely punished, and (2) the well-behaved nodes have slightly worse average ratings. This makes sense since with larger pause times, nodes will naturally interact less among themselves and, therefore, the recovery mechanism which is based on direct interaction will be less effective. This fact resulted in a sharp change in the evolution of the average primary rating when the malicious node stops misbehaving at the 300 seconds, which was not so clear in the previous plots with a pause mean of 1 second, because the recovery mechanism had more impact on the ratings due to larger amount

of interactions between nodes.

5.4 Summary and Conclusions

In this chapter we have evaluated our protocol with respect to both the overhead introduced by our extensions to the regular OLSR operation, and to the effectiveness of the punishment of malicious nodes by the reputation system. For the overhead evaluation, both analytical and simulation-based methods were used, whereas the effectiveness of our reputation system was evaluated through simulation-based scenarios.

As final remarks, since the mechanism for detection of fake HELLO behaves arguably well for the several rates considered, we believe that a CPM rate of 15% would be the wisest choice. In terms of the mechanism for detection of fake TC, a CPM rate in between 15% and 40% would provide a reasonable punishment to the malicious node with ratings around 75-80 points for the weakest attacker (a single faked link) while keeping a low overhead to the network. For a larger number of faked links or a larger pause mean the punishment becomes more severe.

It is important to mention that our reputation scheme presents a simple way of solving common problems of reputation systems. In particular, it does not require the dissemination of reputation information throughout the network. Moreover, nodes are not able to falsely accuse or praise other nodes because either they would have to generate fake CPMs (which can be protected by cryptographic mechanisms such as those used to prevent tampering of routes of on-demand ad-hoc routing in [HPJ05]) or repeat old CPMs (which are protected by a timestamp mechanism).

Chapter 6

Conclusions

We presented a reputation-based security scheme for OLSR, which deals with the generation of fake HELLO and fake TC messages, two attacks which so far did not have a satisfactory solution.

To the best of our knowledge this is the first implementation of a reputation-based security scheme in a standardized proactive routing protocol for MANETs. Moreover, besides providing a natural solution to secure the routing protocol control traffic, our scheme presents a simple way of solving common problem of reputation systems. In particular, it does not require the dissemination of reputation information throughout the network, and nodes are not able to falsely accuse or praise other nodes.

6.1 Future Work

As part of our ongoing work we are studying how to better tackle the bogus detections of misbehavior (so that all well behaved nodes are guaranteed to maintain maximum ratings at all times), how to develop more effective recovery mechanisms, and how to tackle the traffic relay refusal attack. At a more conceptual level, we are aiming at a game theoretic analysis of the proposed approach.

References

[ACJ+03] Cedric Adjih, Thomas Clausen, Philippe Jacquet, Anis Laouiti, Paul Mühlethaler, and Daniele Raffo. Securing the OLSR protocol. In *Proceedings of Med-Hoc-Net*, Mahdia, Tunisia, June 2003.

[ACL+05] Cedric Adjih, Thomas Clausen, Anis Laouiti, Paul Mühlethaler, and Daniele Raffo. Securing the OLSR routing protocol with or without compromised nodes in the network. Technical Report INRIA RR-5494, HIPERCOM Project, INRIA Rocquencourt, February 2005.

[ARM05] Cedric Adjih, Daniele Raffo, and Paul Mühlethaler. Attacks against OLSR: Distributed key management for security. In *2005 OLSR Interop and Workshop*, Ecole Polytechnique, Palaiseau, France, July 28–29 2005.

[BB02] Sonja Buchegger and Jean-Yves Le Boudec. Performance analysis of the confidant protocol. In *MobiHoc '02: Proceedings of the 3rd ACM international symposium on mobile ad hoc networking & computing*, pages 226–236, New York, NY, USA, 2002. ACM Press.

[BCGS04] Stefano Basagni, Marco Conti, Silvia Giordano, and Ivan Stojmenović. *Mobile Ad Hoc Networking*. Wiley-IEEE Press, 2004.

[BH00] Levente Buttyán and Jean-Pierre Hubaux. Enforcing service availability in mobile ad-hoc wans. In *MobiHoc '00: Proceedings of the 1st ACM international symposium on mobile ad hoc networking & computing*, pages 87–96, Piscataway, NJ, USA, 2000. IEEE Press.

[BH03] Levente Buttyán and Jean-Pierre Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *Mobile Networks and Applications*, 8(5):579–592, 2003.

[BMA02] Mounir Benzaid, Pascale Minet, and Khaldoun Al. Integrating fast mobility in the OLSR routing protocol. *4th International Workshop on Mobile and Wireless Communications Network*, pages 217–221, 2002.

[CHCB01] T. Clausen, G. Hansen, L. Christensen, and G. Behrmann. The Optimized Link State Routing Protocol, Evaluation through Experiments and Simulation. *IEEE Symposium on Wireless Personal Mobile Communications*, 2001.

[CJ03] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). Technical report, IETF Internet Draft, http://www.ietf.org/rfc/rfc3626.txt, 2003.

[DRWL04] D. Dhillon, T. S. Randhawa, M. Wang, and L. Lamont. Implementing a fully distributed Certificate Autorithy in an OLSR MANET. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2004)*, Atlanta, Georgia, USA, March 21–25 2004.

[FV06] K. Fall and K. Varadhan. The ns manual. Technical report, The VINT Project. Available from http://www.isi.edu/nsnam/ns/ns-documentation.html, 2006.

[GT00] David Goldsman and Gamze Tokol. Output analysis procedures for computer simulations. In *WSC '00: Proceedings of the 32nd conference on Winter simulation*, pages 39–45, San Diego, CA, USA, 2000. Society for Computer Simulation International.

[HPJ05] Y.C. Hu, A. Perrig, and D.B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. *Wireless Networks*, 11(1):21–38, 2005.

[ID03] Mohammad Ilyas and Richard C. Dorf, editors. *The handbook of ad hoc wireless networks*. CRC Press, Inc., Boca Raton, FL, USA, 2003.

[JLMV01] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot. Performance analysis of OLSR multipoint relay flooding in two ad hoc wireless network models. *INRIA research report RR-4260*, 2001.

[JM96] D.B. Johnson and D.A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 353:153–181, 1996.

[JMC+01] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Vennot. Optimized link state routing protocol for ad hoc networks. In *Proc. of IEEE International Multitopic Conference (INMIC 2001)*, 2001.

[KCC05] Stuart Kurkowski, Tracy Camp, and Michael Colagrosso. MANET simulation studies: the incredibles. *Mobile Computing and Communications Review*, 9(4):50–61, 2005.

[MGLB00] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *MobiCom '00: Proceedings of the 6th annual international conference on mobile computing and networking*, pages 255–265, New York, NY, USA, 2000. ACM Press.

[MM02a] P. Michiardi and R. Molva. Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proc. of the IFIP-Communication and Multimedia Security Conference*, Copenhagen, June 2002.

[MM02b] P. Michiardi and R. Molva. Simulation-based analysis of security exposures in mobile ad hoc networks. *European Wireless Conference*, 2002.

[MM04] C.S.R. Murthy and BS Manoj. *Ad Hoc Wireless Networks: Architectures and Protocols.* Prentice Hall PTR Upper Saddle River, NJ, USA, 2004.

[OTL04] R. Ogier, F. Templin, and M. Lewis. Topology Dissemination Based on Reverse-Path Forwarding (TBRPF). Technical report, IETF Internet Draft, http://www.ietf.org/rfc/rfc3684.txt, 2004.

[PB94] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 234–244, New York, NY, USA, 1994. ACM Press.

[PBV05] Santashil PalChaudhuri, Jean-Yves Le Boudec, and Milan Vojnovic. Perfect simulations for random trip mobility models. In *Annual Simulation Symposium*, pages 72–79. IEEE Computer Society, 2005.

[PR99] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 2, 1999.

[Qay00] A. Qayyum. *Analysis and Evaluation of Channel Access Schemes and Routing Protocols in Wireless LANs.* PhD thesis, Université de Paris-sud, Orsay, France, 2000.

[RACM04] Daniele Raffo, Cedric Adjih, Thomas Clausen, and Paul Mühlethaler. An advanced signature system for OLSR. In *SASN '04: Proceedings of the 2nd ACM Workshop on security of ad hoc and sensor networks*, pages 10–16, New York, NY, USA, 2004. ACM Press.

[Raf05] Daniele Raffo. *Security Schemes for the OLSR Protocol for Ad Hoc Networks.* PhD thesis, Université Paris, 2005.

[RR04] P. Ruiz and F. Ros. UM-OLSR. *Obtain via: http://masimum.dif.um.es/?Software:UM-OLSR*, 2004.

[Sim05] N. Simulator. ns-2. *Obtain via: http://nsnam.isi.edu/nsnam/index.php/Main_Page*, 2005.

[VB06] João P. Vilela and João Barros. A Cooperative Security Scheme for the Optimized Link State Routing in Mobile Ad-hoc Networks. In *15th IST Mobile and Wireless Communications Summit*, Mykonos, Greece, June 2006.

[Vil06] João P. Vilela. CSS-OLSR - Code for simulations, 2006. Obtain via: http://www.ncc.up.pt/~joaovilela/css-olsr/.

[ZCY03] S. Zhong, J. Chen, and Y. R. Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad hoc networks. In *Proceedings of INFOCOM*, San Francisco, USA, March 2003.