Securing OLSR Using Node Locations

Daniele Raffo Cédric Adjih Thomas Clausen Paul Mühlethaler

INRIA Rocquencourt, HIPERCOM project
Domaine de Voluceau, B.P.105, 78153 Le Chesnay cedex, France
{Daniele.Raffo, Cedric.Adjih, Paul.Muhlethaler}@inria.fr, T.Clausen@computer.org

***Abstract*: In this paper we examine security issues related to the Optimized Link State Routing protocol, a proactive routing protocol for MANETs. We enumerate a number of possible attacks against the integrity of the OLSR routing infrastructure, and present a technique for securing the network. In particular, we concentrate on the remaining attacks when a mechanism of digitally signed routing messages is deployed and an attacker may have taken control over trusted nodes. Our solution is based on inclusion of the geographical position of the sending node in control messages, and on evaluation of verisimilitude of links; this is accomplished using a GPS device and a directional antenna embedded in each node.**

1. Introduction

A Mobile Ad hoc NETwork (MANET) is a collection of nodes that are able to connect on a wireless medium to form an arbitrary and dynamic network. Implicit here is the characteristic of the network topology to change over time as links in the network appear and disappear.

In order to enable communication between any two nodes in such a MANET, a routing protocol is employed. The abstract task of the routing protocol is to discover the topology (and, as the the network is dynamic with continuing changes to the topology) to ensure that each node is able to acquire a recent view of the network topology in order to construct routes.

Currently, two complementary classes of routing protocols exist in the MANET world. Reactive protocols acquire routes on demand (this class includes protocols such as AODV [22] and DSR [14]), while proactive protocols ensure that topological information is maintained through periodic message exchange (this class includes OLSR [7], OSPF [18], DSDV [23], and TBRPF [20]).

1.1. Security Issues and Related Work

A significant issue in the ad hoc domain is that of the integrity of the network itself. Routing protocols allow, according to their specifications, any node to participate in the network, with the assumption that all nodes are trusted and following the protocol. If malicious nodes inject wrong control messages in the network, the integrity of the network fails.

The primary issue with respect to securing MANET routing protocols is thus of ensuring network integrity, even in the presence of malicious nodes. Security extensions to the reactive protocols AODV and DSR exist, in the form of SAODV [28] and Ariadne [10] respectively. SAODV uses digital signatures on Route Request and Route Reply messages. Ariadne authenticates the sender by using clock synchronization and delayed key disclosure. Another reactive protocol, ARAN [26], uses an authenticated route discovery. Regarding the proactive protocols OLSR and OSPF, a system of digital signatures has been proposed [1, 24, 19]. The secured version of DSDV, named SEAD [9], uses hash chains for message authentication.

Maintaining the integrity of the network becomes more difficult when an intruder has taken control of a trusted node (which then becomes a malicious node) or has captured its private key; the intruder then is able to send authenticated messages. Known security techniques against this kind of attack which aim at identifying and blacklisting the faulty nodes, are the Watchdog/Pathrater [17], CONFIDANT [3] and WATCHERS [2, 12].

In this paper we will investigate the issues of security in the OLSR proactive protocol, with emphasis on providing an improved security extension. We will introduce a mechanism, inspired by the work of Hu et al. about packet leashes [11], to ensure network integrity even under the assumption of nodes fallen under the control of an attacker. Such a property did not exist with the security architecture proposed in a previous paper [1].

1.2. Paper Outline

The remainder of this paper is organized as follows: Section 2. gives an overview of the OLSR protocol. Section 3. describes the vulnerabilities of proactive routing protocols, using OLSR to exemplify the threats to which any proactive ad hoc routing protocol is vulnerable.

Section 4. presents a security solution we have proposed for OLSR, and which uses digital signatures. This is used as a starting point for our new location-based solution described in Section 5.. Finally, Section 6. concludes the paper.

2. The OLSR Protocol

The Optimized Link State Routing protocol (OLSR) [7, 13, 6] is a proactive link state routing protocol for mobile ad hoc networks. OLSR employs an optimized flooding mechanism for diffusing link state information, and diffuses only partial link state to all nodes in the network.

In this section we will describe the elements of OLSR required for the purpose of investigating security issues.

2.1. OLSR Control Traffic

Control traffic in OLSR is exchanged through two different types of messages: HELLO and TC messages.

HELLO messages are emitted periodically by a node and contain three lists: a list of neighbors from which control traffic has been heard, a list of neighbor nodes with which bidirectional communication has been established, and a list of neighbor nodes that have been selected to act as Multipoint Relay for the originator of the HELLO message. HELLO messages are exchanged between neighbor nodes only, and are not forwarded further.

Upon receiving a HELLO message, a node examines the lists of addresses. If its own address is included in the addresses encoded in the HELLO message, bidirectional communication is possible between the originator and the recipient of the HELLO message, i.e. the node itself.

In addition to information about neighbor nodes, periodic exchange of HELLO messages allows each node to maintain information describing the links between its neighbor nodes and nodes which are two hops away. This information is recorded in a nodes 2-hop neighbor set and is utilized for MPR optimization – see section 2.2..

Like HELLO messages, TC messages are emitted periodically. The purpose of a TC message is to diffuse link state information to the entire network. Thus, a TC message contains a set of bi-directional links between a node and a subset of its neighbors. TC messages are flooded into the entire network, taking advantage of the MPR optimization described in section 2.2.. Only the nodes which have been selected as an MPR generate TC messages.

An individual OLSR control message can be identified by its “Originator Address” and “Message Sequence Number” – both from the message header. Hence it is possible to uniquely refer to a specific control message in the network.

2.2. Multipoint Relay Selection

The core optimization in OLSR is that of *Multipoint Relays* (MPRs) [25, 5]. Each node must select MPRs from among its symmetric neighbor nodes such that a message emitted by a node and repeated by the MPR nodes will be received by all nodes two hops away. In fact, in order to achieve a network-wide broadcast, a broadcast transmission needs only to be repeated by just a subset of the neighbors: this subset is the MPR set of the node. Hence only MPR nodes relay control messages.

Figure 1 shows the node in the center, with neighbors and 2-hop neighbors, broadcasting a message. In (a) all nodes retransmit the broadcast, while in (b) only the MPRs of the central node retransmit the broadcast.

Each node maintains a *MPR selector set*, describing the set of nodes which have selected it as an MPR.

3. Vulnerabilities

We discuss now various security risks in OLSR. The aim is not to emphasize flaws in OLSR, as it did not include security measures in its design, like several other routing protocols; while these vulnerabilities are specific to OLSR, they can be seen as instances of what other link state routing protocols are subject to.

Under a proactive routing protocol, each node must correctly generate routing control traffic conforming to the specification, and forward routing control traffic on behalf of other nodes. An intruder can carry out attacks against the routing protocol aimed at impeding the formation of the network, making legitimate nodes store incorrect routes, and more generally perturbing the network topology. An attack may also have the aim of modifying the routing protocol, so that traffic flows through a specific node controlled by the intruder. Often, the intruder will first need to gain full control over (to *compromise*) a legitimate node, which then will start misbehaving. *Denial of service* attacks against the physical layer (e.g. jamming, radio interference, etc.) are not discussed in this paper.

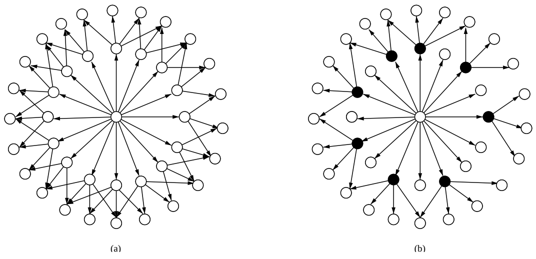

Figure 1: Pure flooding (a) and MPR optimized flooding (b).

3.1. Incorrect Traffic Generation

3.1.1. Identity spoofing

Identity spoofing implies that a misbehaving node sends control messages pretending to be another node. A misbehaving node X may send HELLO messages with a spoofed originator address set to that of node A (Figure 2). Subsequently, nodes B and C may announce reachability to A through their HELLO and TC messages. Furthermore, node X chooses MPRs from among its neighbors, signaling this selection while pretending to have the identity of node A. Therefore, the chosen MPRs will advertise in their TC messages that they provide a “last hop” to A. Conflicting routes to node A, with possible connectivity loss, may result from this.

TC messages with a spoofed originator address cause incorrect neighbor relationship to be advertised in the network. For instance, node X sends a TC message on behalf of node A, advertising C as a neighbor. The other nodes, upon reception of the TC message, will falsely conclude that A and C are neighbors.

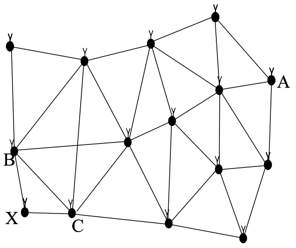

Figure 2: Identity spoofing: node X sends HELLOs with the identity of node A.

3.1.2. Link spoofing

We call link spoofing the signaling of an incorrect set of neighbors in a control message, i.e. the signalisation of neighbor relationship with non-neighbor nodes. A misbehaving node X may perform link spoofing in its HELLO messages, advertising a link with non-neighbor node A, as in Figure 3. This will result in X's neighbors storing an incorrect 2-hop neighborhood, therefore selecting a wrong MPR set. Node X can also misbehave by signaling an incomplete set of neighbors. Depending on their links with other nodes, the ignored neighbors might experience breakdown in connectivity with the rest of the network.

TC messages with spoofed links have the same effect, and can severely perturb the network topology as stored by legitimate nodes. Node X, behaving incorrectly, can also send TC messages without being an MPR. The protocol specification states that only MPRs generate TC; however, there are no ways of detecting whether the originator of a TC message is a MPR of some node or not.

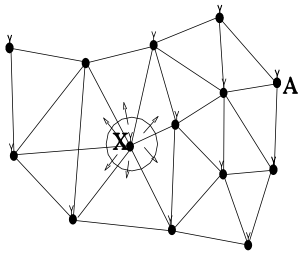

Figure 3: Link spoofing: node X advertises an inexistent link with A.

3.2. Incorrect Traffic Relaying

3.2.1. Failure in relaying control packets

If a node fails to relay TC messages, the network may experience connectivity problems. In networks where no redundancy exists (e.g. in a "strip" network), connectivity loss will surely result, while other topologies may provide redundant connectivity.

3.2.2. Replay attack

As topology changes, old control messages, while valid in the past, describe a topology configuration not existing anymore. An attacker can resend old valid control messages (which were previously recorded by the attacker), to make other nodes update their routing tables with stale routes. This is known as a replay attack, and is successful even if control messages bear a digital signature that is not timestamped.

3.2.3. Wormhole attack

This severe attack [11] consists of recording traffic from one region of the network and replaying it in a different region. It is carried out by an intruder node X located within transmission range of legitimate nodes A and B, where A and B are not within transmission range with each other. Intruder node X merely tunnels control traffic between A and B (and vice versa), without the modification presumed by the routing protocol – e.g. without stating its address as source in the packets header – so that X is virtually invisible. This results in an extraneous inexistent $A - B$ link which in fact is controlled by X. Node X can then drop tunneled packets or break this link at its will. Two intruder nodes X and Y, connected by a wireless or wired private medium, can also collude to create a longer (and more harmful) wormhole, as shown in Figure 4.

Marshall points out a similar attack [16] against the Secure Routing Protocol [21].

The severity of the attack comes from the fact that it is difficult to detect, and is effective even in a network where authentication, integrity, and confidentiality are preserved (e.g. via encryption and digital signatures). Furthermore, on a distance vector routing protocol, wormholes are very likely to be chosen as routes because they provide a shorter path (albeit compromised) to the destination.

In OLSR, to successfully exploit the wormhole, the attacker must wait until A and B have exchanged sufficient HELLO messages (through the wormhole) to establish a symmetric link. Until that moment, other tunneled control messages would be rejected, because the OLSR protocol specifies that TC/MID/HNA messages must not be processed if the relayer node (the "last hop") is not a symmetric neighbor. However, once created, the $A - B$ link is at the mercy of the attacker.

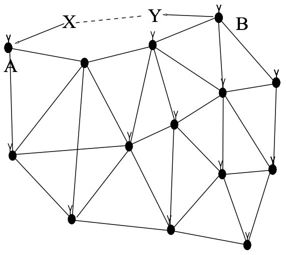

Figure 4: Wormhole attack: intruder nodes X and Y create an artificial link between A and B.

4. Previous Studies: Security Solutions

4.1. Overview

In a previous paper [1], we proposed a mechanism to secure the OLSR protocol by signing and timestamping control messages. A new kind of control message (SIGNATURE) is sent along with any HELLO and TC, and contains the signature of the HELLO/TC as well as a timestamp. The signature is computed on the sequence of bits made from all the fields of the HELLO/TC message and all the fields of the SIGNATURE message (except of course the "Signature" field itself). The mechanism requires a PKI and a timestamp synchronization algorithm between the nodes.

4.2. Protection Offered

The signature and timestamping mechanisms protect network integrity against attacks such as incorrect traffic generation and incorrect traffic relaying. The timestamp avoids the occurrence of replay attacks. Since an intruder node cannot sign valid control messages properly, such messages will be dropped when their signature is verified by any receiving node. As a consequence, an intruder will not be able to participate in the network.

However, this security architecture [1] is unable to repel wormhole attacks. Furthermore, this architecture is unable to protect the network if a legitimate node is compromised, i.e. if a legitimate node has fallen under the control of an attacker, or its private key has been disclosed to the attacker in any way. In both cases, the attacker has access to the node's private key, and can steal the node's identity by sending messages signed on behalf of that node.

To block these attacks, we have designed a solution which consists of including the node's geographical position as additional information in the SIGNATURE message.

5. Adding Node Location in Signatures

5.1. Overview

The attacks shown in the previous section can be thwarted if we possess *node position information*, that is, if every node is able to know the correct geographical position of any other node in the network. Nodes then compare this geographical data to the received routing data (i.e. the neighbor and link set). If contradictory information is found, the false routing message is detected and discarded.

The geographical position can be obtained by using Global Positioning System (GPS) devices embedded into the hardware of each node. [1] There exist other solutions which do not require every node to be equipped with a GPS device [27] or which do not use GPS at all [4]. However, due to the possible presence of malicious nodes, solutions which rely on feedback or signals from other nodes (e.g. the emission power) cannot be considered safe.

An additional security measure is obtained by using a directional antennae instead of an omni [8]. This allows a node to know the direction from which a received message was transmitted, and therefore makes it much more difficult for malicious nodes to spoof their own location.

Besides, the availability of geographical information about nodes in the network opens speculations about possible new features in the standard OLSR, such as improved MPR selection and link breaking forecast. However, completely replacing the basic link state discovery mechanism based on HELLOs/TCs with a position-based mechanism is not desirable. The reason is that the position of nodes gives a mere estimation of their link probability; adjacent nodes may not be able to communicate due to an obstacle, etc. These issues are not within the scope of this paper, and are not discussed further.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sign. Method  |   Reserved    |        MSN Referrer           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Node location                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Timestamp                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                          Signature                            :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: SIGLOC message format.

5.2. Specifications

We therefore suggest some modifications to the security protocol [1] already proposed for OLSR. A SIGLOC (which stands for SIGnature and LOCalization) control message substitutes the SIGNATURE message; the former includes a new "Node Location" field, which contains the current geographical position of the sending node as obtained from the GPS facility. This field is 32 bits long (which is enough to define the position over an area of more than 4200 square km with a granularity of 1 m), and is included in the signature computation. The message format is given in Figure 5. This mechanism requires the deployment of a Public Key Infrastructure and a timestamp synchronization algorithm between all nodes. These topics are not discussed further in this paper; please refer to the solution detailed in the original security scheme [1].

A node informs the other nodes about its current position in a SIGLOC message (which, we recall, is sent along with every generated HELLO and TC).[2] The receiving node first couples the SIGLOC with its companion HELLO/TC (by matching the SIGLOC's MSN Referrer with the HELLO/TC's Message Sequence Number) and verifies the correctness of the timestamp and signature, as already specified by the original security protocol [1]; then it extracts the position information and stores the tuple $\langle$ node address, position, timestamp $\rangle$ in a *position table*. For each node, the most recent positions are memorized in the position table.

The advantage in knowing the geographical position of nodes is that a receiver node can speculate whether or not a link is likely to exist. This link may be a direct link with a neighbor or a link advertised in a TC message.

5.2.1. Checking the originator of a HELLO message

We call p_A the current position of the receiver A, and t_A the time (according to A's clock) when it receives a control message from node B (Figure 6). Node A learns in the SIGLOC message the position p_B of node B at time t_B (according to B's clock). We call Δt the discrepancy in the clocks synchronization of the nodes, Δd the maximum absolute error in position information, v

[1] The same GPS facility can be used to provide time synchronization e.g. up to 20 ns. [15].

[2] For improved security, it is possible to generate SIGLOC messages also in conjunction to MID/HNA messages. MID and HNA messages have, in OLSR, respectively the purpose of declaring multiple interfaces of a node, and host associations with non-OLSR networks.

the maximum velocity of any node in the network, and r the maximum transmission range. Taking into account errors and nodes motion, $\|p_A - p_B\|$ must satisfy the following:

$$\|p_A - p_B\| \leq r + (t_A - t_B + \Delta t) \cdot 2v + 2\Delta d = r + 2i \quad (1)$$

When formula 1 is not valid, this means that the receiver node is too far from the sender node to be able to hear its transmission. Therefore such an HELLO message is highly suspicious and might well be tunneled as a wormhole attack. The receiver should drop such HELLO message.

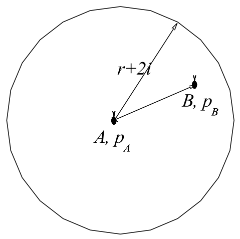

Figure 6: Test of likelihood when a HELLO message is received.

When a directional antenna is used, the receiver node knows which direction the signal is coming from. Basing on the $p_B - p_A$ value and using simple geometry, this allows the receiver node to check roughly the correctness of the position p_B declared by the sender node (Figure 7). We denote with $\overrightarrow{AB}$ the vector linking node A and node B, as established with the directional antenna, when the HELLO message is received (at time t_A); then the current relative position of B, from node A, should be within a sphere of diameter $2i$ with center in $\overrightarrow{p_B - p_A}$, i.e.

$$\|\overrightarrow{AB} - (p_B - p_A)\| \leq \ (t_A - t_B + \Delta t) \cdot 2v + 2\Delta d = 2i \quad (2)$$

This information can be useful if the transmission range r is not known with precision. In such a case we can set a lower bound on the transmission range and derive from formula 2 the sector in which the sender should be. Should the directional antenna indicate another direction for the reception, such a transmission must be considered as a fake, and the receiver should drop such a HELLO message.

If we assume that a node velocity is linked to the validity time given in the OLSR packet, we can refine formulas 1 and 2 with a better estimation of the velocity of A and B.

5.2.2. Checking links advertised in HELLO and TC messages

Formula 1 also permits to detect and reject the false control messages described in section 3.1.2.. Node C receives a control message sent by node A, advertising a link between node A and node B, as shown in figure

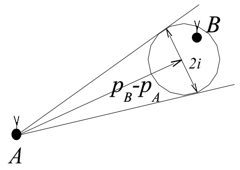

Figure 7: Test of likelihood when a HELLO message is received using a directional antenna.

8. The position of the originator node A is found in the SIGLOC message to be p_A at time t_A. The value p_B is the location of the node B at time t_B, as found in node C's location table. The actual location should be found for a given time t'_B that minimizes $|t_A - t'_B|$ (Figure 8). An interpolation for the position of B can be used if the location of node B is not known for a time close to t_A.

If formula 1 is not satisfied, then the $A - B$ link is suspicious and the control message advertising this link must be dropped.

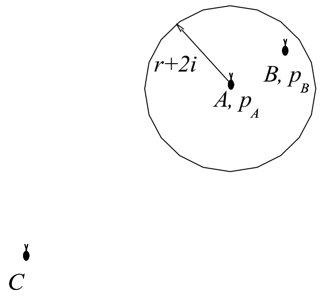

Figure 8: Test of likelihood for links advertised in HELLO or TC messages.

5.3. Procedure when Creating a Control Message

When node A generates a HELLO or a TC, it must also generate a SIGLOC and perform the following steps:

1. create the SIGLOC message
2. write the node position
3. write the timestamp of the actual time
4. compute and write the signature of the HELLO/TC message
5. send the HELLO/TC and the SIGLOC

5.4. Procedure when Receiving a Control Message

When a node receives a HELLO or TC originating from A, the following actions must be carried out:

1. pair off correctly the HELLO/TC with its SIGLOC companion, by matching the Message Sequence Number with the MSN Referrer

2. check the freshness of the timestamp

3. check the validity of the signature

4. check the validity of the HELLO message with respect to its originator node (according to Section 5.2.1.), and the validity of links advertised in a HELLO/TC message (according to Section 5.2.2.).

5. store the tuple $\langle$ address of A, node location, timestamp $\rangle$ in the position table

If any of these tests fail, the HELLO/TC and the SIGLOC must be immediately dropped.

5.5. Protection Offered

Simple signatures with timestamps are sufficient to thwart attacks such as incorrect traffic generation and incorrect traffic relaying, under the assumption that only legitimate nodes can sign control packets. Adding the node location in the signature message allows the network to avoid wormhole attacks and false messages due to compromisation of nodes.

5.5.1. Repelling wormhole attacks

We analyze the consequences of formula 1. With figures such as $v = 60$ km/h, $t_A - t_B + \Delta t \leq$ 100 msec and $\Delta d = 1$ meter, the originator node should be within a radius of $r + 5.333$ meter. When r is not too small (e.g $r > 50$ meter) , the control packet received is necessarily sent by a nearby node (within the coverage of the recipient). This means that wormhole attacks tunneling such a control packet would be difficult to launch because the real control packet is likely to be heard by the recipient; on the other hand, such an attack would be not very efficient, since the node whose control message is relayed will be, most likely, at most two hops away. When r is small (e.g. 20 meter $< r <$ 50 meter) the information given by a directional antenna can be useful, since the sector in which the signal is expected has a limited size.

Note that a wormhole attack tunneling a TC message through two distant points of the network may not be repelled, since the signature carries the position of the originator node, and the originator of a TC message may not be within reach. Furthermore, as OLSR mandates that a TC message is not relayed when it has already been received from a node which is not a MPR selector, the correctness of MPR flooding can be affected by this attack. This could be avoided by signing the whole OLSR packet hop by hop. However, this would introduce other difficulties, as the content of a packet may change hop by hop: TC messages are relayed while HELLOs are not.

5.5.2. Repelling incorrect traffic generation

Formula 1 also permits to reject control messages advertising impossible links because the two endpoints are too far from each other. Therefore, the SIGLOC signature is efficient in repelling such link spoofing attacks.

5.6. Overhead

We can mathematically evaluate the overhead increase caused by the sending of SIGLOC messages. The size of a HELLO message advertising n nodes can be averaged as $136 + 40n$, while the size of a TC message advertising n neighbors is $32(n + 1)$ bits.

We assume the use of a 128-bit signature, and a 32-bit timestamp, which is enough to define the time value for a period of more than 49 days with a granularity of 1 ms. The resulting size of a SIGLOC message will be 224 bits.

A SIGLOC message is generated and sent with every HELLO or TC. By assuming an average neighborhood of 12 nodes, this will result in an overhead increase of about 136% for each HELLO message, and about 154% for each TC message, with respect to the standard OLSR protocol. These evaluations do not include the size of OLSR, IP and UDP packet headers.

There is also an overhead in terms of the time required for signature computation and verification, which is not evaluated in this paper as it is implementation-dependent.

6. Conclusion

In this paper we have examined some major issues related to the security of the OLSR proactive link state protocol. We have enumerated a number of possible attacks against OLSR, and stressed those attacks which can be carried out against a network where the authentication and integrity of messages are insured by digital signatures. We then proposed a solution which relies on adding the geographical position of a node into control messages. This can be obtained by embedding GPS devices and directional antennas in the nodes' hardware. The implementation of this solution is proposed as an extension to a digital signature infrastructure which we presented in a previous paper. As some of the insights provided are general to a larger class of link state protocols, the proposed solution may offer hints toward making these protocols more secure.

REFERENCES

[1] 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 25–27 2003.

[2] Kirk A. Bradley, Steven Cheung, Nick Puketza, Biswanath Mukherjee, and Ronald A. Olsson. Detecting disruptive routers: A distributed network monitoring approach. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 1998.

[3] Sonja Buchegger and Jean-Yves Le Boudec. Performance analysis of the CONFIDANT protocol (Cooperation Of Nodes: Fairness In Dynamic Adhoc NeTworks). In *Proceedings of MOBIHOC*, EPFL Lausanne, Switzerland, June 9–11 2002.

[4] Srdan Capkun, Maher Hamdi, and Jean-Pierre Hubaux. GPS-free positioning in mobile ad hoc networks. In *HICSS*, 2001.

[5] Thomas Clausen, Philippe Jacquet, and Laurent Viennot. Investigating the impact of partial topology in proactive MANET routing protocols. In *Proceeding of Wireless Personal Multimedia Communications*. MindPass Center for Distributed Systems, Aalborg University and Project Hipercom, INRIA Rocquencourt, Fifth International Symposium on Wireless Personal Multimedia Communications, November 2002.

[6] Thomas Heide Clausen, Gitte Hansen, Lars Christensen, and Gerd Behrmann. The Optimized Link State Routing protocol, evaluation through experiments and simulation. In *Proceedings of the IEEE Symposium on Wireless Personal Mobile Communications*, September 2001.

[7] T. Clausen (ed) and P. Jacquet (ed). Optimized link state routing protocol (OLSR), October 2003. RFC 3626, Experimental.

[8] Rob Flickenger. *Building Wireless Community Networks*. O'Reilly & Associates Inc., 2003.

[9] Yih-Chun Hu, David B. Johnson, and Adrian Perrig. SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002)*, pages 3–13, Calicoon, NY, USA, June 2002.

[10] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *Proceedings of the 8th ACM International Conference on Mobile Computing and Networking*, September 2002.

[11] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, San Francisco, CA, USA, April 2003.

[12] John R. Hughes, Tuomas Aura, and Matt Bishop. Using conservation of flow as a security mechanism in network protocols. In *IEEE Symposium on Security and Privacy*, pages 131–132, 2000.

[13] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. IEEE INMIC, 2001. Hipercom Project, INRIA Rocquencourt.

[14] David B. Johnson, David A. Maltz, and Yih-Chun Hu. The dynamic source routing protocol for mobile ad hoc networks (DSR), February 24 2003. Internet-Draft, draft-ietf-manet-dsr-08.txt.

[15] Trimble Navigation Limited. Data sheet and specifications for thunderbolt GPS disciplined clock, 2000. `http://www.trimble.com`.

[16] John Marshall. An analysis of SRP for mobile ad hoc networks. In *Proceedings of the 2002 International Multiconference in Computer Science*, Las Vegas, USA, August 18–21 2002.

[17] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Mobile Computing and Networking*, pages 255–265, 2000.

[18] J. Moy. OSPF version 2, April 1998. RFC 2328, Standards Track.

[19] S. Murphy, M. Badger, and B. Wellington. OSPF with digital signatures, June 1997. RFC 2154, Experimental.

[20] R. Ogier, F. Templin, and M. Lewis. Topology dissemination based on reverse-path forwarding (TBRPF), February 2004. RFC 3684, Experimental.

[21] Panagiotis Papadimitratos and Zygmunt J. Haas. Secure routing for mobile ad hoc networks.

[22] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing, July 2003. RFC 3561, Experimental.

[23] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244. ACM Press, 1994.

[24] Ricardo Staciarini Puttini, Ludovic Me, and Rafael Timóteo de Sousa. Certification and authentication services for securing manet routing protocols. In *Proceedings of the Fifth IFIP TC6 International Conference on Mobile and Wireless Communications Networks*, Singapore, October 2003.

[25] Amir Qayyum, Laurent Viennot, and Anis Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical report, Hipercom Project, INRIA Rocquencourt, 2000. INRIA RR-3898.

[26] Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth M. Belding-Royer. A secure routing protocol for ad hoc networks. In *Proceedings of the 10th IEEE International Conference on Network Protocols*, pages 78–89. IEEE Computer Society, 2002.

[27] Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179. ACM Press, 2001.

[28] Manel Guerrero Zapata. Secure ad hoc on-demand distance vector (SAODV) routing, October 2002. Internet-Draft, draft-guerrero-manet-saodv-00.txt.