OLSR with GPS information

Daniele Raffo Cedric Adjih Thomas Clausen Paul Mühlethaler

Daniele.Raffo@inria.fr Cedric.Adjih@inria.fr T.Clausen@computer.org Paul.Muhlethaler@inria.fr

INRIA Rocquencourt, Hipercom project
Domaine de Voluceau, B.P.105, 78153 Le Chesnay cedex, France
Telephone: +33 1 3963 5363 Fax: +33 1 3963 5566

***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 or may not have full control over trusted nodes. Our solution is based on the inclusion of the geographical position, obtained by a GPS device, of the sending node in control messages. This solution may also be applied to other link state protocols.

I. Introduction

A Mobile Ad hoc NETwork (MANET) is a collection of nodes which are able to connect on a wireless medium to form an arbitrary and dynamic network. Implicit herein 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, continuing changes to the topology) to ensure that each node is able to acquire a recent image of the network topology for constructing routes.

Currently, two complimentary classes of routing protocols exist in the MANET world. Reactive protocols acquire routes on demand (this class includes protocols such as AODV [18] and DSR [12]), while proactive protocols ensure that topological information is maintained through periodic message exchange (this class includes OLSR [6], OSPF [15], and TBRPF [17]).

A. 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 that assumption fails - i.e. the network is subject to malicious nodes - the integrity of the network fails.

The primary issue with respect to securing MANET routing protocols is thus that of ensuring network integrity, even in the presence of malicious nodes. Security extensions to the reactive protocols AODV and DSR exist, respectively in the form of SAODV [22] and Ariadne [8]. SAODV uses digital signatures on the Route Request and Route Reply messages. Ariadne authenticates the sender by using clock synchronization and delayed key disclosure. A system of digital signatures for the proactive protocols OLSR and OSPF has been proposed, respectively in [1] and [16]; see also [19].

Maintaining the integrity of the network becomes more difficult when an intruder has compromised a trusted node (which hence becomes a malicious node) or has captured its private key; the intruder then becomes able to send authenticated messages. Known security techniques against this kind of attack are the Watchdog/Pathrater [14], CONFIDANT [3] and WATCHERS [2], [10], which aim at identifying and blacklisting the faulty nodes.

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 to ensure network integrity and detect misbehaving nodes.

B. Paper outline

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

Section IV presents a security solution which we have proposed for OLSR, and which uses digital signatures. This is used as a starting point for our new GPS-based solution described in section V. Finally, section VI concludes the paper.

II. The OLSR Protocol

The Optimized Link State Routing protocol (OLSR) [11], [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.

A. 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 MPR 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, bi-directional 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 II-B.

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 to the entire network, exploiting the MPR optimization described in section II-B. Only 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. This will become important when discussing message signatures.

B. Multipoint Relay Selection

The core optimization in OLSR is that of *Multipoint Relays* (MPRs). Each node must select MPRs from among its 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. MPR selection is performed based on the 2-hop neighbor set received through the exchange of HELLO messages, and is signaled through the same mechanism.

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

Figure 1 shows a node with neighbors and 2-hop neighbors. In order to achieve a network-wide broadcast, a broadcast transmission needs only be repeated by just a subset of the neighbors. This subset is the MPR set of the node.

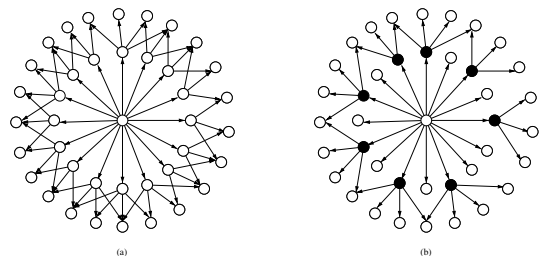

Fig. 1. Two hop neighbors and Multipoint Relays (the solid circles) of a node. (a) All neighbors retransmit a broadcast. (b) Only the MPRs of a node retransmit the broadcast.

Further details on OLSR are discussed in [6], [20], [5].

III. Vulnerabilities

In this section, we discuss various security risks in OLSR. While these vulnerabilities are specific to OLSR, they can be seen as instances of what all proactive 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. By carrying out an attack against the routing protocol, an intruder can perturb or paralyze the whole network. Often, the intruder will first need to gain full control of a trusted node, which then will start misbehaving. In the rest of this section we will show how routing misbehavior may appear in OLSR. *Denial of service* attacks against the physical layer (e.g. jamming, radio interference, etc.) are not discussed in this paper.

A. Incorrect Traffic Generation

1) Identity spoofing: Identity spoofing implies that a misbehaving node sends control messages pretending to be another node. Node X sends HELLO messages, with the originator address set to that of node A, as illustrated in figure 2. This may result in the network containing conflicting routes to node A. Specifically, node X will choose MPRs from among its neighbors, signaling this

selection while pretending to have the identity of node A. The MPRs will, subsequently, advertise in their TC messages that they can provide a "last hop" to node A. Conflicting routes to node A, with possible loops, may result from this. Similarly, TC messages with a spoofed originator address cause incorrect links to be advertised in the network.

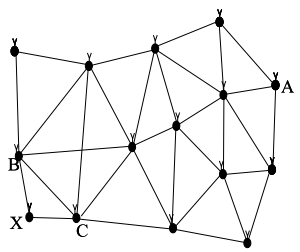

Fig. 2. Identity spoofing: node X sends HELLOs with the identity of node A. As a consequence, nodes B and C may mistakenly announce reachability to A through their TCs.

2) Link spoofing: Link spoofing implies that a node sends control messages signaling an incorrect set of neighbors. A misbehaving node advertising in its HELLO messages a neighbor relationship to non-neighbor nodes may cause inaccurate MPR selection, with the result that some nodes may not be reachable in the network. Again, TC messages which include non-existing links may result in routing loops and conflicting routes in the network (figure 3).

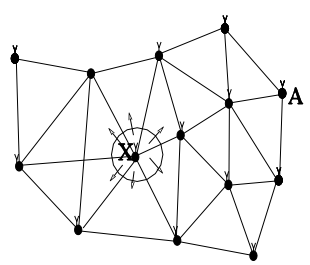

Fig. 3. Link spoofing: node X generates incorrect TCs advertising a link with A.

A node may also misbehave by signaling an incomplete set of neighbors, which might therefore lead to a breakdown in connectivity with the rest of the network.

B. Incorrect Traffic Relaying

1) Failure in relaying: If TC messages are not properly relayed 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.

2) Wormhole attack: A wormhole attack [9] is a severe attack in which traffic from one region of the network is recorded and replayed (selectively or not) in a different region. This attack is effective even if no node has been compromised, and even if all communications are authenticated (e.g. via digital signatures) and confidentiality is preserved (e.g. via encryption).

As regards OLSR, an attacker may use an intruder node which is in the neighborhood of both A and B to relay HELLO messages from A to B and viceversa. The intruder may also perform this attack by recording a message from A, moving quickly into the neighborhood of B and replaying the message there. In the OLSR protocol, where links are discovered by testing reception, this will result in extraneous link creation (a "short" wormhole) between A and B. Alternatively, the attacker may use two intruder nodes, one in the neighborhood of A and the other in the neighborhood of a distant node Z, connected via a direct wireless or wired carrier; the attacker may then tunnel HELLO messages through this longer carrier to create an extraneous A–Z link (a "long" wormhole). The consequences of this attack is that nodes store an incorrect topology of the network.

IV. Previous Studies: Security Solutions

A. Overview

In [1] we proposed a mechanism to secure the OLSR protocol, by signing control messages. This is done by assigning a private/public key pair to every node. A new kind of control message (SIGNATURE_MESSAGE) is sent along with any HELLO and TC, and contains the signature of the HELLO/TC as well as a timestamp (to thwart replay attacks). 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 SIGNATURE_MESSAGE contains a "MSN Referrer" field that holds the same value of the HELLO/TC's "Message Sequence Number" [6] it is coupled with; this allows a node to correctly couple a received SIGNATURE_MESSAGE with a received HELLO/TC. Upon reception, the node then verifies the freshness of the timestamp and the correctness of the signature, and if both are verified, the node processes the message; otherwise the message is dropped. The mechanism requires a PKI and a timestamp synchronization algorithm between the nodes.

B. Protection Offered

The signature mechanism protects the network against the injection of false routing messages by an intruder, and

against any case of identity spoofing (explained in section III-A.1). This is because the intruder does not know the private key of another node, and routing messages which are not properly signed will always be rejected. However, this solution is far from perfect. Wormhole attacks can still be carried out over a network where message authentication and integrity are insured. Furthermore, if the attacker gains control of a node, he/she will have the possibility to generate correctly signed, but nonetheless false, messages.

To block these attacks, we have designed a solution which consists in including additional information into control messages, which is the node's geographical position. Our solution is implemented on the basis of [1], and the node's position is contained in a new field of the modified SIGNATURE_MESSAGE.

V. OLSR WITH GPS

A. Overview

The attacks shown in the previous section can be thwarted if we possess *node position information*, i.e. 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 [21] 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 can be obtained by using directional antennae instead of omni [7]. This will allow 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.

B. Specifications

We suggest therefore some modifications to the protocol illustrated in [1]. A SIGLOC (which stands for SIGnature and LOCalization) control message substitutes the SIGNATURE_MESSAGE; the former includes a new field "GPS Localization", 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 4. 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 [1] for more details.

```
 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          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        GPS Localization                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                           Signature                           :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 4. SIGLOC message format.

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). The receiving node first couples the SIGLOC with its companion HELLO/TC and verifies the correctness of the timestamp and signature, as specified by the protocol in [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 position is memorized in the position table.

The advantage in knowing the geographical position of nodes is that a receiver node can speculate whether communications from a sending node are likely to be heard or not. Let p_r be the current position of the receiver, and t_r the current time according to the receiver's clock. Also let Δt be the discrepancy in the clocks' synchronization, Δd the maximum absolute error in position information, v the maximum velocity of any node, and r the maximum transmission range. Based (from the SIGLOC message) on the position p_s of the sender node and the timestamp t_s, the receiver node can compute a lower bound on the distance d_{sr} between the sender and itself. In fact it must be

$$r \geq d_{sr} \geq \|p_r - p_s\| - (t_r - t_s + \Delta t) \cdot 2v - \Delta d \quad (1)$$

When (1) is not valid, it means that the receiver node is too far from the sender node to be able to hear its transmission, therefore such a transmission is highly suspicious and might well be a fake. Furthermore, when directional antennae are used, the receiver node can know from which

[1] The same GPS facility can be used to provide time synchronization; see [13] for an example.

direction the signal is coming. Based on p_r and using simple geometry, this allows the receiver node to check roughly the correctness of the position p_s declared by the sender node.

C. Protection offered

1) Protection against wormhole attacks: When a message is being maliciously tunneled between legitimate nodes A and B, as described in section III-B.2, (1) is not valid with respect to the distance d_{AB}. Therefore B notices that the transmission is likely being tunneled through a wormhole and should drop the message.

2) Protection against link spoofing: The equation (1) also allows the case of false routing messages, described in section III-A.2, to be detected. For any communication between a sender and a receiver, (1) must hold valid and this obviously also applies to links. We can therefore detect the case in which a misbehaving node X falsely advertises a link (in a HELLO message) with the non-neighbor node N, or declares N as a neighbor (in a TC message). This is done as follows: node A checks in its position table the location of node N advertised in X's HELLO or TC; if (1) is not valid with respect to the distance d_{XN}, A should drop the message.

If X is able to tamper with the GPS Localization field, i.e. X is able to declare that it transmits from another location, then it could declare a location in transmission range of non-neighbor node B and then falsely advertise a link with B. In this case, (1) is valid with respect to the distance d_{XB}. This would severely perturb the network topology if B is very far from X; however, in this case such an attack will not be successful, because node A which receives X's HELLO/TC will notice that (1) is not valid with respect to the distance d_{XA}.

D. The protocol

We detail here the protocols for the creation or reception of control messages.

When node A generates a HELLO or TC, it must also generate a SIGLOC by following this protocol:

1) create the HELLO/TC and the SIGLOC
2) insert the GPS Localization from the GPS device output
3) insert the Timestamp from the actual time
4) compute the Signature on the HELLO/TC + SIGLOC
5) send the HELLO/TC and the SIGLOC

When a node receives a control message from A, it must follow this protocol:

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 with A's public key
4) if using a directional antenna, then check the congruity of GPS Localization with the direction the transmission came
5) for each Neighbor Address I listed in the HELLO/TC: if I is in the position table, check the validity of (1) on d_{AI}
6) store the tuple $\langle$ address of A, GPS Localization, Timestamp $\rangle$ in the position table

If any of the tests fail, the node must not further process the two messages (HELLO/TC and SIGLOC) and must drop them.

E. Overhead

We can mathematically evaluate the overhead increase caused by the sending of SIGLOC messages. The size of a HELLO message advertising n nodes varies from $32(n+2)$ to $32(2n+1)$ bits, depending whether the nodes have the same link/neighbor status or not. The size of a TC message advertising n neighbors is $32(n+1)$ bits.

For performance reasons we choose an Elliptic Curve Cryptography algorithm for signing, hence the use of ECDSA with a 320-bit signature. We also assume the use of 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 416 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 52% to 93% for each HELLO message, and an overhead increase of 100% 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.

VI. Conclusions

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 in the nodes' hardware. The implementation of this solution is proposed as an extension to a digital signature infrastructure that 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 towards making this protocol more secure.

REFERENCES

[1] Cedric Adjih, Thomas Clausen, Philippe Jacquet, Anis Laouiti, Paul Mhlethaler, 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 Ad-hoc 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] T. Clausen (ed) and P. Jacquet (ed). Optimized link state routing protocol (OLSR), October 2003. RFC 3626, Experimental.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

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

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

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

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

[19] 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.

[20] 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.

[21] 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.

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