Reputation Based Intrusion Detection System

Rahul Ambekar, Piyush Shukla, Pushkar Vadkhalkar, Anushree Warang, Sandesh Rane

Abstract -- The unique characteristics of mobile ad hoc networks, such as shared wireless channels, dynamic topologies and a reliance on cooperative behaviour, makes routing protocols employed by these networks more vulnerable to attacks than those employed within traditional wired networks. We consider the security of control traffic generated by pro-active or table-driven link state protocols in mobile ad-hoc networks. Focusing on the Optimized Link State Routing (OLSR) protocol, we propose a specification-based intrusion-detection model for ad hoc routing protocols in which network nodes are monitored for operations that violate their intended behaviour and rewards nodes depending on their cooperation in the exchange of routing information. We design a detection mechanism based on finite state automata for checking whether a network node violates the constraints &correlates direct observation of transmissions with path information from successfully delivered packets.

Index Terms - CSS, Hello message, MPR's, misbehaviour, node, OLSR, TC message.

1. Introduction

The popularity of powerful new wireless technologies has given rise to several new applications. Many of these applications are designed to deploy mobile ad-hoc networks (MANETs) in various environments that include cellular phone services, disaster relief, emergency services, and battlefield scenarios, among others. MANETs are particularly attractive since they enable a group of mobile nodes to communicate using the wireless medium in the absence of pre-existing infrastructure such as base stations.

It is now widely accepted that the specific cooperation mechanisms of MANETs are a source of additional vulnerabilities thus requiring novel security solutions beyond those of the infrastructure /wired paradigm. In the absence of a fixed infrastructure that establishes a line of defence by identifying and isolating non-trusted nodes, it is possible that control messages generated by routing protocols, e.g. neighbour advertisements or link state data, are corrupted or compromised thus jeopardizing the communication within the network.

- *Mr. Rahul Ambekar is a professor and project guide in PVPPCOE, Mumbai, India.*
- *The other mentioned authors are currently pursuing a Bachelors degree in Engineering (IT) in PVPPCOE (Class of 2012) University of Mumbai, India.*

E-mail:piyusl04@yahoo.co.in
E-mail:pushkar_1710@yahoo.com

Among the numerous proposals for routing protocols in MANETs, the Optimized Link State Routing (OLSR) protocol is arguably one that offers promising performance in terms of bandwidth, required overhead and delivered traffic albeit at the cost of a wide range of security challenges, mostly with respect to the required exchange of topology information and the underlying design assumption that all nodes are benign.

The goal of this paper is to provide the OLSR protocol with a security solution that defends the network against malicious

nodes by monitoring of the individual nodes for violation of the constraints developed from the correct behaviour of the

nodes & rewarding proper routing behaviour and thus assuring effective cooperation between communicating parties. After introducing MANET in Section 1, we give a brief account of OLSR protocol along with its control messages in Section 2.Moving over to Section 3 we highlight the existing system including the Specification Based Architecture and an FSA model for the same. We throw light on CSS (Cooperative security scheme) and discuss its techniques in detail. Thereafter we suggest a proposed system in Section 4 and move on to the acknowledgements, conclusion and references.

Although our analysis of these security issues, which includes a thorough review of related work, is mainly focused on the OLSR protocol, the described problems and the proposed solutions are equally applicable to other common routing protocols for MANETs.

2. Optimized Link State Routing (OLSR)

OLSR is a proactive table-driven link-state routing protocol developed by INRIA. As a proactive routing protocol, it has the advantage of making the routes immediately available

when needed, and as a link state protocol, it uses flooded information about the network topology to calculate the best next-hop for every possible destination in the network. OLSR defines a term MPRs Multi-Point Relays as the 1- hop neighbour of a node that is used to reach 2-hop neighbours.

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

• *Reduction of the size of control packets* by declaring only a subset of links with its neighbours who are its *multipoint relay selectors* (MPR selectors);

• *Minimization of flooding* by using only a set of selected nodes, called *multipoint relays* (MPRs), to diffuse its messages to the network (only the multipoint relays of a node retransmit its broadcast messages).

The use of MPR's for message transmission results in a scoped flooding instead of full node-to-node flooding thus inducing a reduction in the amount of exchanged control traffic. The protocol is particularly suitable for large and dense networks, because the optimization procedure based on multipoint relays works best in those cases.

2.1 There are two types of control messages in OLSR

1) HELLO messages are periodically broadcasted by each node, containing its own address, neighbour lists and the corresponding link state for each of them (unidirectional, bi-directional or MPR). These messages are only exchanged between neighbouring nodes but they allow each node to have information about one and two-hop neighbours 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 neighbours who have selected the sender node as a MPR (MPR selector set) and a sequence number associated to the MPR selector set.

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 neighbour nodes that will retransmit its packets. This set of nodes is called the *multipoint relay set* of that node and can change over time, as indicated by the selector nodes in their HELLO messages. The node which chooses the multipoint relay set is a *multipoint relay 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 neighbours covering all the two-hop neighbours. Additionally, all two hop neighbours must have a bi-directional link to the selected MPR set. The smaller the multipoint relay set, the more efficient the routing protocol.

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 broadcasts traffic control information about the one-hop neighbours that selected it as a multipoint relay (or, equivalently, its multipoint relay 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 multipoint relays from the source to the destination. The neighbours of any node which are not in its MPR set receive and process the control traffic but do not retransmit it.

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 neighbours but are not relayed;
3) HELLO messages provide each node with knowledge about one and two-hop neighbours;
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.

In a proactive routing protocol, each node has two tasks to accomplish:

(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).

Table 1. Critical fields in Hello and TC Messages

Message Type	Critical fields
Hello Message	1-hop neighbour list MPR sets
TC Message	MPR selectors Advertised neighbour sequence number (ANSN)

Thus, an attack on the routing protocol must result as the corruption of one of these tasks by some node. Thus, an attack can:
1. Provide an incorrect 1-hop neighbour list in a Hello message
2. Provide an incorrect MPR set in a Hello message
3. Provide incorrect MPR selectors in a TC message
4. Modify the MPR selectors before it forwards a TC message

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 (as he is expected).

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.

3. Existing Solutions

In this section we provide the introduction of two available security solutions and describe their approach.

3.1 Specification Based Intrusion Detection System

Intrusion detection is a viable approach to enhancing the security of existing computers and networks. Briefly, an intrusion detection system monitors activity in a system or network in order to identify on going attacks. Intrusion detection techniques can be classified into anomaly detection, signature-based detection, and specification-based detection. In anomaly detection, activities that deviate from the normal behaviour profiles, usually statistical, are flagged as attacks. Signature-based detection matches current activity of a system against a set of attack signatures. Specification-based detection identifies system operations that are different from the correct behaviour model.

The specification-based approach [1] analyzes the protocol specification (e.g., RFC) of an ad hoc routing protocol to establish a finite-state-automata (FSA) model that captures the correct behaviour of nodes supporting the protocol. Then, it extracts the constraints on the behaviour of nodes from the FSA model. Thus, the approach reduces the intrusion detection problem to monitoring of the individual nodes for violation of the constraints. Such monitoring can be performed in a decentralized fashion by cooperative distributed detectors, which allows for scalability. In addition, since the constraints are developed based on the correct behaviour, this approach can detect both known and unknown attacks.

In general, specification-based detection recognizes attacks by comparing the activity of an object with a model of correct behaviour of the object. It has been applied to detect attacks on computer programs and network protocols. Specification-based detection is particularly suitable for detecting attacks on network protocols because the correct behaviour of a protocol is well defined and is documented in the protocol specification. The challenge of this approach is to extract a suitable correct behaviour model from the protocol specification that can be checked at runtime using network monitoring.

Thus after the analysis of the OLSR RFC the following assumptions were listed for the application of this approach:

- We assume a distributed intrusion detection architecture that allows cooperative detectors to promiscuously monitor all Hello and TC messages, and exchange their local data if necessary.
- IDS detectors in this architecture can monitor all Hello and TC messages sent by each node of the network, always exchange IDS data successfully, and will not be compromised.
- In addition, we assume that cryptographic protection, such as TESLA, is employed to guard against spoofing attacks.
- Furthermore, we assume OLSR is the only routing protocol in the network and each node has only one network interface. In other words, Multiple Interface

Declaration (MID) and Host and Network Association (HNA) messages are not used here.

- Lastly, we assume nodes forward TC messages following OLSR Default Forwarding Algorithm and nodes forward normal packets to the correct next hop.

3.2 Correct Behaviour Model of OLSR

The following figure shows the FSA model of the OLSR protocol that defines the correct operation of an OLSR node in handling control traffic.

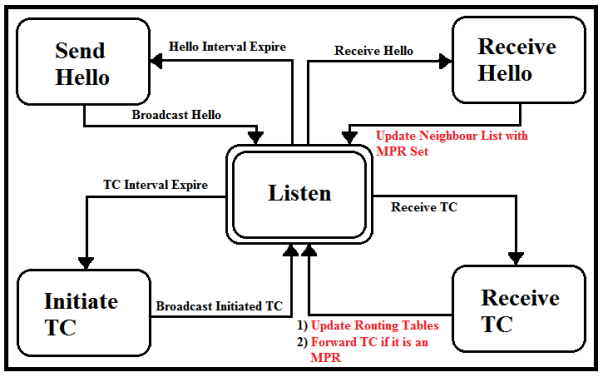

Fig. OLSR Routing Finite State Automata (FSA)

The constraints on the control traffic between neighbour nodes for detecting inconsistencies within the control messages:
C1: Neighbour lists in Hello messages must be reciprocal.
E.g., if node 2 is the neighbour of node 1, then node 1 must be node 2's neighbour.

C2: The MPR nodes of a node must reach all 2-hop neighbours of the node and the MPR nodes must transmit TC messages periodically.

C3: MPR selectors of a TC message must match corresponding MPR sets of Hello messages.
E.g., if node 2 is node 1's MPR selector, node 1 must be in node 2's MPR set.

C4: Fidelity of forwarded TC messages must be maintained.

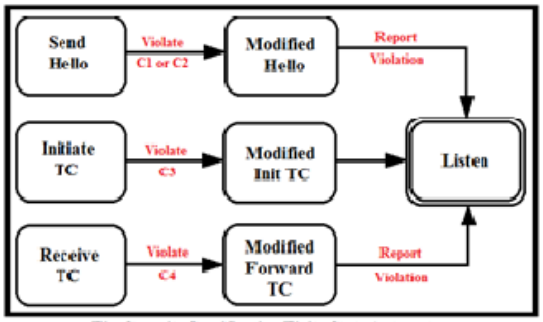

Fig. Security Specification Finite State Automata

Implementation of Detection Mechanism
The proof-of-concept prototype is implemented as a global detector that can monitor all Hello and TC messages in the simulated OLSR network. It is important to note that although the current prototype is a centralized detector, the proposed intrusion detection model can be implemented in a decentralized fashion.

As the goal of the proof-of-concept prototype is to validate the detection model, a centralized implementation suffices for validating the false positive and false negative characteristics under our assumptions.

Four data tables are maintained by the global detector to record 1-hop neighbours, 2- hop neighbours, MPR and MPR selector sets of all nodes. Four constraints are evaluated according to data tables and incoming messages. An alert will be raised if a constraint is violated. However, topology changes will cause temporal inconsistency and lead to false alert. To minimize the false positive rate, we develop a mechanism to detect temporal inconsistency between new message and old history data. First, the threshold time is set for each constraint according to intervals of Hello messages and TC message. Then it generates alerts only when an inconsistency last beyond the threshold time of a constraint.

The constraints are coded inside the main OLSR protocol class and are executed in recurring fashion.

Pseudo code of Constraint C1 :
Constraint 1 (1-hop Table, node i)
For each 1-hop neighbour j in 1-hop Table i
If i is not in 1-hop Table j // if there is inconsistency between link states of node i and j
{ If 1-hop Table(i,j).alert = = FALSE //if no inconsistency before
{Set 1-hop Table(i,j).alertTime = Current Time //set time stamp
Set 1-hop Table(i,j).alert = TRUE //mark the inconsistency}

Else {If (Current Time - 1-hop Table(i,j). alertTime) > Threshold of C1 // if inconsistency
Raise Alarm of C1}} //has lasted more than threshold, raise an alarm
Else{1-hop Table(i,j).alert = FALSE}

3.3 Co-operative Security Scheme

The fundamental concern behind the proposed Cooperative Security Scheme [2] for OLSR (CSS-OLSR) is that of assuring that nodes correctly generate and relay OLSR control traffic. The scheme is based on rewarding nodes that cooperate with the routing protocol to tackle two security issues: fabrication of false routing messages and traffic relay refusal. To achieve this goal, the guiding principles will be to reward well behaved nodes and to strongly penalize damaging behaviour.

This scheme adds three 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 node sends a CPM back to the originator with the path traversed by the original TC message which, therefore, must have recorded the path traversed by itself (e.g. by setting the record route flag in the IP header or keeping the information on the payload of a TC message).

• *Rating table:* Each node of the network keeps a rating table which holds information about the behaviour of its one and two-hop neighbours. Each entry in the rating table has a node ID, a primary and a secondary rating. The node ID uniquely identifies a node, the secondary rating is a classification of a node based on the direct observation, and the primary rating is a more mature classification of a node based on its secondary rating and the matching of the information provided by CPMs with the information announced by a node. The information maintained on this table enables the nodes to decide how to handle misbehaving nodes.

• *Warning message:* Another type of messages called potential misbehaviour warning message is used to notify neighbour nodes of potential misbehaviour of nodes.

Since CSS-OLSR requires the ability to identify each node and the exact origin of each packet, it relies on the use of a distributed CA's.

3.3.1. Protocol Specification

A security extension to the OLSR protocol that employs the proposed scheme can be defined as follows.
i) At the formation of the network, a distributed CA is employed guarantying the proper authentication of each node;
ii) Each time a new node enters the network, the distributed CA is used to ensure the node's authenticity;
iii) During the broadcast of HELLO messages to ensure knowledge of one and two-hop neighbours, only properly authenticated nodes are considered;
iv) For each authenticated node found, a new entry in the rating table is added with value α for the secondary rating and $\varrho$for the primary rating;
v) The same as items 4, 5, 6 and 7 of the original OLSR protocol (as described in Sec.2);
vi) 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;
vii) The same as items 8 and 9 of the original OLSR protocol (as described in Sec.2).

3.3.2. Detection of misbehaviour through direct observation

The detection of misbehaving through direct observation is done by having each node to listen promiscuously to its MPR transmissions. If the source node of a communication, *S*, detects that a MPR did not relay its message, it decreases the MPR secondary rating by $\tau 1$ and sends a potential misbehaving message to all one-hop neighbours. Upon receipt of this message, each neighbour of *S* decrements the MPR secondary rating by $\tau 2$. Otherwise, if the MPR is detected to relay the message, its secondary rating is increased by γ, but only by node *S*.

To encourage cooperation, the punishment should be greater than the reward, i.e. $\tau 1 > \gamma$. Additionally, the fact that only the source node *S* increases the secondary rating by direct observation and all of its one-hop neighbours decrease it if the node misbehaves makes it harder for a node to keep a good reputation and misbehave often.

In order to motivate nodes to behave well, a node A relays node B's traffic based on the primary rating of B in A, specifically the primary rating controls the rate at which node A relays node B traffic.

3.3.3. Detection of misbehaviour through analysis of the CPMs

Although OLSR assumes a bidirectional connection between a node and its MPRs, in the following scenarios a node may not

detect misbehaviour through direct observation of its neighbours: packet collisions, limited transmission power, nodes collusion and partial packet dropping. Therefore the secondary rating (obtained through direct observation of other node's packet forwarding) is only used as an unreliable node status. To classify nodes as misbehaving the primary rating is used. The primary rating is obtained through correlation of the secondary rating and information gained from the CPMs.

To prevent redundant information to be used, upon the reception of a CPM by a node, say node *A*, if the CPM has a path that *A* has sent to his neighbours within a certain period of time β, or a packet generated by the same node has been received within the same period of time, *A* discards it. Otherwise, the processing is as specified in the following Algorithm 1.

Algorithm 1 CPM processing

1: SR_{MPR}←secondary rating of the MPR in *A*'s rating table
2: PR_{MPR}←primary rating of the MPR in *A*'s rating table
3: **if** *A* is the intended receiver of the CPM and *A* has sent a TC message to the network within a short period of time δ **then**
4: **if** the information in the CPM is consistent with the information obtained from the MPR by *A* **then**
5: **if** $SR_{MPR}>PR_{MPR}$ **then**
6: PR_{MPR}←SR_{MPR}
7: **else**
8: SR_{MPR}←$SR_{MPR}+\gamma$
9: **end if**
10: **else**
11: **if** $SR_{MPR}<PR_{MPR}$ **then**
12: PR_{MPR}←SR_{MPR}
13: **else**
14: SR_{MPR}←$SR_{MPR}-\tau 1$
15: **end if**
16: **end if**
17: *A* forwards the CPM to all one-hop neighbours.
18: **else**
19: Forward the CPM as usual.
20: **end if**

Basically, Algorithm 1 states that if node *A* is the intended receiver of the CPM and has sent a TC message within a period of time δ(step 3), *A* finds the MPR to which he forwarded the packet, say M1, and checks (a) if the hop after M1 in the path contained in the CPM belongs to the MPRs of M1 and (b) if that hop is the one expected by the current routing table of *A*.

If so, and if the secondary rating of M1 is bigger than the primary rating of M1 (which corresponds to the node being well behaving), the primary rating of M1 gets the value of the secondary rating of the same node (step 6).

If the secondary rating is lower than the primary rating (the node has been reported as misbehaving) the information of the secondary rating might be corrupted (because direct observation of nodes forwarding is error-prone) and the secondary rating is increased by γ (step 9).

Otherwise, if the information in the CPM is not consistent with what M1 advertises (step 11) and the secondary rating of M1 is lower than the primary (misbehaving node), the primary rating of M1 is set to the value of the secondary rating (step 13). If the secondary rating is bigger than the primary M1 seems to be well behaving, but because the (more important) CPM information shows the opposite, M1 secondary rating is decreased by $\tau 1$ (step 15). Afterwards, *A* forwards the packet to all one-hop neighbours for the same processing. At each node we only verify if its own MPRs are behaving correctly (generating correct traffic and relaying traffic that is sent to them). Although, as the proposed changes in CSS-OLSR are distributed in the sense that every node in the network executes them, the tampering of a message somewhere along a path will also be detected and punished, eventually not by the source node of the message, but by closer nodes in the path.

4. Proposed System

We propose a security solution that combines both the above mentioned approach and utilizes their benefits in a collective and collaborative manner.

In our approach we follow the follow the following algorithm:

- Define four new attributes in original OLSR namely Global Detectors, CPM, Rating Table, Warning Message.
- Define the FSA constraint model and bind it as an internal function in the OLSR routing protocol file.
- Define the CPM and Rating Table processing as an explicit function.
- Follow the CSS-OLSR protocol specification (as mention in section 3.3) and for and after every topology update run the global detector function.

The proposed system is under the implementation phase and is being carried out in Simulation environment with the help of Network Simulator 3.

The proposed system will provide the following benefits over the existing system:

- It combines the FSA model along with the CPM feedback which allows operation in the correct manner and along with increased reliability and availability.
- The overall overhead is minimal in comparison to either of the two approaches when implemented alone.
- The system assures network integrity and also helps detect and prevent both Active as well as Passive attacks.

5. Conclusion

The main goal of this paper is to present the combinational model and the benefits of both the security solutions which can be combined and implemented for large networks where integrity and authentication are of primary concern.

Analysing the OLSR routing specification, we define the normal OLSR routing behaviour and list possible attack mechanisms from a single attacker. Based on the normal routing behaviour, nodes retrieve routing information, and establish and maintain their routing tables correctly using the Hello and TC messages. We develop constraints on these Hello and TC messages in order to establish that the integrity of the routing tables at all nodes is not compromised. We develop the proof of satisfaction of the requirement that the integrity of routing tables of all nodes is safeguarded.

CSS-OLSR inherits the benefits of distributed certificate authorities enabling it to identify each node and the exact origin of each packet without a centralized approach. This way, identity spoofing attacks are addressed and countered, whereas to defend against replay attacks the traditional usage of timestamp mechanisms can be relied upon. Beyond these well-understood aspects, our scheme, which correlates error-prone information obtained through direct observation of node transmissions with information obtained from the paths traversed by successfully delivered packets, along with constant monitoring with the help of intrusion detection system.

6. Acknowledgment

We sincerely acknowledge the effort and the belief put into our project by our Principal Mr K.T.V Reddy, HOD Prof. Vaibhav Narawade, Project Guide Prof. Rahul Ambekar and Project Co-ordinator Prof. Langewar.

References

[1] Chinyang Henry Tseng, Poornima Balasubramanyam, "*Specification Based Intrusion Detection Model for OLSR*".

[2] Joao P. Vilela, Joao Barros, "*A Cooperative Security Scheme for Optimized Link State Routing in Mobile Ad-Hoc Networks*".

[3] Frederic Cuppens, "*Misbehaviour Detection to ensure Availability in OLSR*".

[4] Cedric Adijh, "*Attacks against OLSR: Distributed Key Management Security*".

[5] Frank Kargl, "*Advanced detection of Selfish or Malicious Nodes in Ad Hoc Networks*".

[6] L Premaajeshwari, "*Enhanced Intrusion Detection Techniques for mobile Ad Hoc networks*".

[7] T Clausen and P Jacquet, "*Optimized State Link Routing Protocol IETF RFC 3626*".

[8] Cedric Adijh, T Clausen, P. Jacquet, "*Securing the OLSR protocol*", http:// www.olsr.org

[9] Jo˜ao P. Vilela Jo˜ao Barros, *"A Cooperative Security Scheme for Optimized Link State Routing in Mobile Ad-hoc Networks".*

[10] Jo˜ao P. Vilela Jo˜ao Barros, *"A Feedback Reputation Mechanism to Secure the Optimized Link State Routing Protocol".*

[11] D. Raffo, *"Security schemes for the OLSR protocol for ad hoc networks"* Ph.D. dissertation, Universit´e Paris, 2005.

[12] T. Clausen and P. Jacquet *"Optimized link state routing protocol (olsr), rfc 3626"* October 2003.