Attack and Defence Evaluation of Security Methods in Mobile Ad hoc Networks

JIMMI GRÖNKVIST, ANDERS HANSSON, DAN NORDQVIST, MATTIAS SKÖLD

FOI, Swedish Defence Research Agency, is a mainly assignment-funded agency under the Ministry of Defence. The core activities are research, method and technology development, as well as studies conducted in the interests of Swedish defence and the safety and security of society. The organisation employs approximately 1250 personnel of whom about 900 are scientists. This makes FOI Sweden's largest research institute. FOI gives its customers access to leading-edge expertise in a large number of fields such as security policy studies, defence and security related analyses, the assessment of various types of threat, systems for control and management of crises, protection against and management of hazardous substances, IT security and the potential offered by new sensors.

FOI
Swedish Defence Research Agency
Command and Control Systems
P.O. Box 1165
SE-581 11 Linköping

Phone: +46 13 37 80 00
Fax: +46 13 37 81 00

www.foi.se

FOI-R--2215--SE
ISSN 1650-1942

Technical Report
December 2006

Command and Control Systems

Attack and Defence Evaluation of Security Methods in Mobile Ad hoc Networks

FOI-R–2215–SE
ISSN 1650-1942

Command and Control Systems
Technical Report

December 2006

Issuing organization Swedish Defence Research Agency Command and Control Systems P.O. Box 1165 SE-581 11 LINKÖPING SWEDEN	**Report number, ISRN** FOI-R–2215–SE	**Report type** Technical Report
	Programme areas 4. C^4ISR	
	Month year December 2006	**Project No.** E7075
	Subcategories 41. C^4I	
	Subcategories 2	
Author/s Jimmi Grönkvist, Anders Hansson, Dan Nordqvist, and Mattias Sköld	**Project manager** Jimmi Grönkvist	
	Approved by Sören Eriksson	
	Sponsoring agency Swedish Armed Forces	
	Scientifically and technically responsible Lars Ahlin	
Report title Attack and Defence Evaluation of Security Methods in Mobile Ad hoc Networks		

Abstract

A mobile ad hoc network consists of wireless nodes that build a robust radio network without any pre-existing infrastructure or centralized servers. However, these networks have inherent vulnerabilities that make them susceptible to malicious attacks. In order to secure ad hoc networks advanced techniques must be used, one efficient solution is to use specification-based intrusion detection, especially when combined with traditional cryptographic methods.

In this report, we have studied attacks on realistic networks to see what effect they have on communications. We show that some of the well known attacks on AODV do have a significant effect, preventing more or less all nodes from communicating. However, as we also show, our specification-based Intrusion Detection System removes almost all of the effects of the attacks by discarding detected incorrect packets. This can be done with very little cost in terms of overhead and false alarms. We have also studied OLSR and show that the same methods could be applied also on this protocol.

Keywords

Ad hoc networks, AODV, OLSR, secure routing

Further bibliographic information	**Language** English
ISSN 1650-1942	**Pages** 43 p.
	Price acc. to pricelist

Utgivare Totalförsvarets Forskningsinstitut Ledningssystem Box 1165 SE-581 11 LINKÖPING	**Rapportnummer, ISRN** FOI-R–2215–SE	**Klassificering** Teknisk rapport
	Forskningsområde 4. Spaning och ledning	
	Månad, år December 2006	**Projektnummer** E7075
	Delområde 41. Ledning med samband och telekom och IT-system	
	Delområde 2	
Författare Jimmi Grönkvist, Anders Hansson, Dan Nordqvist, och Mattias Sköld	**Projektledare** Jimmi Grönkvist	
	Godkänd av Sören Eriksson	
	Uppdragsgivare/kundbeteckning Försvarsmakten	
	Teknisk och/eller vetenskapligt ansvarig Lars Ahlin	
Rapportens titel Utvärdering av attacker och försvar av säkerhetsmetoder i mobila ad hoc-nät		
Nyckelord ad hoc-nät, AODV, OLSR, säker routing		
Övriga bibliografiska uppgifter	**Språk** Engelska	
ISSN 1650-1942	**Antal sidor:** 43 s.	
Distribution enligt missiv	**Pris: Enligt prislista**	

Sammanfattning

Ett mobilt trådlöst ad hoc-nät består av ett antal noder, som bildar ett robust radionätverk utan fast infrastruktur och centraliserade funktioner. Dock är dessa nätverk sårbara för nya klasser av attacker t ex denial of service och spridning av falsk routinginformation. För att erhålla acceptabel nivå på säkerheten behövs avancerade säkerhetslösningar. En effektiv variant är att använda policy-baserad intrångsdetektering, speciellt när den kombineras med traditionella kryptolösningar

I den här rapporten har vi studerat attacker på realistiska ad hoc nät för att undersöka hur stor effekt de faktiskt har på kommunikationen. Vi visar att flera av de kända attackerna kan i stort sett slå ut all kommunikation. Genom att ignorera felaktiga paket kan vår policy-baserade intrångsdetekteringsmetod ta bort i stort sett all effekt av dessa attacker, till priset av en liten kostnad uttryckt i minskad tillgänglig kapacitet vid normal drift.

Vi visar också att motsvarande metoder kan användas för OLSR.

Contents

Chapter 1

Introduction

1.1 Ad Hoc Networks

An ad hoc network is a collection of wireless mobile nodes that dynamically form a temporary network without the need for any pre-existing network infrastructure or centralized administration. Due to the limited transmission range of radio interfaces, multiple "hops" may be needed for one node to exchange data across the network with another node. An ad hoc network is both self-forming and self-healing and it can thus be deployed with minimal or no need for network pre-planning. Although one drawback is that the network will not always be connected. A tactical network may therefore be partitioned or fragmented into parts due to e.g., movements or terrain obstacles. It is therfore necessary that parts of the network can function autonomously, and this requires a distributed network control.

The above properties makes ad hoc networks a good choice for tactical military scenarios which can be unpredictable and where the loss of any node is possible. However, the basic properties of ad hoc networks also makes them very difficult to secure. Unlike traditional networks there are no central points which can be used to control access to the network and its resources. Furthermore, all nodes are mobile and are sensitive to hijacking (at more or less degree at least) and the use of radio means that an hostile node can attempt access the network anywhere, which makes it difficult to separate the network into secured and unsecured parts. For example, there is no place for a single firewall or

intrusion detection system that can protect the network.

The traditional method of protecting radio networks have been the use of cryptographic mechanisms, such as encryption and message authentication. However, such mechanisms can only protect against attacks against external nodes and cannot protect against compromised nodes that already are a part of the network, and therefore already may have many of the keys [1].

With the development of Software-Defined Radio and Network Centric Warfare many of the security issues known from Internet may be a possible reality. It is difficult to design and implement software systems without introducing design and programming errors that an adversary can exploit. If an adversary has adequate resources and tries hard enough, there is always a risk that the adversary succeeds in infiltrating the system.

History has taught us that no matter how many security mechanisms (e.g. encryption, authentication and firewalls) that are inserted in the network, there are always weak links that adversaries can exploit.

Hence, to obtain an acceptable level of security in military contexts, traditional security solutions should be coupled with intrusion detection systems (IDS) that continuously monitor the network and determine whether the system (the network or any node of the network) is under attack. Once an intrusion is detected, e.g. in the early stage of a denial of service attack, a response can be put into place to minimize the damage.

1.2 Intrusion Detection Systems

Intrusion detection can be classified into three broad categories [2]: anomaly detection, misuse (signature) detection, and specification-based detection. Anomaly detection recognizes deviations from normalcy by building models of normal behaviour. Any deviation from normal is identified as an attack. Misuse detection use patterns of known attacks to recognize intrusions. Specification-based detection detects attacks with use of a set of constraints (rules) that define the correct operation of a program or a protocol.

Misuse detection has high detection accuracy and low false alarm rate for known attacks, but it is unable to detect novel attacks whose signatures are unknown. The ability to detect previously unknown attacks is essential in military contexts, since some military organizations have resources to develop specific

targeted attacks that are unknown and not used in civilian contexts. Anomaly detection is able to detect unknown attacks. However, anomaly detection techniques also produce a high degree of false alarms, which is not acceptable for an intrusion detection system for a military network. Specification-based detection is similar to anomaly detection in that it is able to detect unknown attacks. The main advantage of specification-based methods is that it provides the capability to detect previously unknown attacks, while providing a low false positive rate, i.e., few false alarms. However, the specifications are usually derived manually from RFCs or other descriptions of protocols. Thus, an obvious downside is that the development of specifications may be time-consuming and protocol specific.

Given the complementary nature of the strengths and weaknesses of prevention approaches, such as authentication, and intrusion detection approaches, a natural approach is to combine the two approaches in such a way that we can realize the combination of their strength, while avoiding the weaknesses of either one.

1.3 Previous Work Within the Project

This report is the final report in a three year long project that have been studying security issues regarding ad hoc network. In the first year the project concentrated on the investigation on general weaknesses in ad hoc networks, presented in [3], and studies different ways to can protect the ad hoc networks. Intrusion detection is seen as a very important component in this protection.

In the second year, the work resulted in a report [4], in which we argue that specification-based intrusion detection combined with traditional cryptographic methods can be one possible way of resolving many security issues regarding ad hoc networks. A specific algorithm was developed for the Ad hoc On-demand Distance Vector (AODV) Protocol and some initial results show positive results.

In this report we finalize the work from the previous years by carry out a more complete evaluation of the efficiency of our intrusion detection algorithm. We will show that the method is still very accurate even in mobile scenarios and that the added overhead does not have a significant effect on network performance.

We also briefly show that the method is applicable for other protocols by

describing a specification-based model for the Optimised Link State Routing Protocol (OLSR).

1.4 Related Work

Most protocols for ad hoc networks have been developed without any consideration for security, assuming that all nodes trust each other; specifically we can mention here the few ad hoc network protocols that have reached the standardization phase, i.e. AODV [5], DSR [6], OLSR [7], and TBRPF [8]. The few protocols for ad hoc networks developed with consideration for security can rather be seen as patches to the existing protocols, e.g. ARIADNE (for DSR) [9] and SAODV (for AODV) [10].

In this report we have chosen to focus our evaluation on AODV. An attempt to secure this protocol have been done in SAODV, by letting each node sign the routing messages with a private key, thus preventing nodes from sending most false messages. In addition, when a node replies to a route request it also adds a signature from the destination to prove that it has a route. Finally, hash chains are used in order to protect the part of a message that intermediate nodes will change on each hop.

However, even if SAODV can protect against several attacks it still has problems. One problem is a node that floods the network with large numbers of route requests but still correctly signs these messages. Another problem is connected to the key management, the present suggestion [10] is that each node picks an asymmetric key pair and from this deterministically generates an IP address. This does not only give huge overhead when the public keys need to be sent over the network to all nodes, but that transmission can also be attacked. There is also the additional problem with mapping an IP address to an identity. Which nodes should be allowed into the network?

Although, these protocols give a considerable improvement to security as compared to the original protocols, their implementation is difficult and there are still unresolved security issues. Pure intrusion prevention methods will not be sufficient in mobile ad hoc networks, therefore during the last few years the research community has put considerable attention to the area of intrusion detection.

One approach to solve this is to generalize intrusion prevention in order to

make networks more resilient towards networks attacks. In [11] an architecture called Techniques for Intrusion-Resistant Ad Hoc Routing Algorithms (TIARA) is suggested, using firewalls, distributed traffic policing and other methods minimizing the effect of a malfunctioning or malicious node. However, this requires a major modification of existing algorithms, which makes it unlikely to be used in the near future.

One of the first papers on intrusion detection for ad hoc networks was published in the year 2000 [12]. In that paper, a basic architecture was described. The approach is based on IDS agents on every node that each performs local detection on local data. Global detection can be initiated when a node reports an anomaly. How the actual detection and verification should be handled was not addressed though.

In [13] the above architecture was used and expanded specifically for AODV. Cooperation between the nodes was assumed and a threshold for malicious nodes was defined, if the threshold is passed a node can initiate a response, and if two or more nodes respond to the same node that node can be purged from the network.

In [14] a so-called watchdog approach is described. Here the nodes study their neighbors to see if they are relaying packets correctly. For example, after a node sends a routing packet it will listen on the channel to see whether that neighbor retransmit the packet correctly without incorrect modifications within a specified time. If a node fails to do this within a certain time the node is marked as misbehaving, such a node is then avoided when a path is chosen.

The problem with this solution is that nodes do not necessarily receive their retransmitted packets. For example, this is the case when nodes are using power control and the next hop requires less power than the first hop. However, it may also have problems in fixed power transmissions since packets may be lost due to collisions and some medium access control protocols, e.g. STDMA, does not guarantee correct reception on any node but the designated receiver. (It may still be a compliment perhaps, since few other methods can determine whether another node randomly drops a packet. How responses should be taken must be considered carefully though since false and missed alarms will be common.)

Several suggestions for using mobile agents also exists [15, 16]. In these protocols mobile agents move around from node to node and collect and study information that have been gathered locally in each node. This has the advantage that the mobile agent can take decisions based on the collected information from

many nodes without having to send all information into a central node.

One problem with these methods is how the mobile agent itself should be protected. If the agent is moved to a malicious node, that node cannot be allowed to change the agent, something that seems difficult in practice.

Traditionally the most used method for IDS in fixed networks have been misuse detection. However, for ad hoc networks most researchers have concentrated on anomaly detection. One motivation is that the data bases for misuse detection will be too expensive for a small ad hoc node, another motivation is that much financing to ad hoc networking research comes from the military sector, in which the ability to detect unknown attacks is essential. As previously mentioned though, anomaly detection gives a large number of false alarms and better methods would be preferable.

For specification-based IDS less work on ad hoc networks have been done. In [17] a finite state machine is used for correcting routing behaviour for AODV, but their solution requires several assumptions that make it unsuited for our scenarios. The most problematic assumption is that they assume that MAC addresses cannot be forged (this is then used to control the origin of a message).

In [10], specification-based IDS is combined with anomaly-based IDS for AODV, but the anomaly part of the IDS incurs a large number of false alarms that makes it difficult to use in our scenarios.

1.5 Overview of the Report

Chapter 2 is an overview of AODV and our specification-based IDS works for this protocol. We also give a description of the specific attacks that we have chosen to implement.

In Chapter 3 we give an overview description of OLSR and describe some of the weaknesses of this protocol. We also describe how an intrusion-based IDS can be developed in this case as well.

Chapter 4 describes scenarios and how simulations are done. Further, we also show the simulation results and motivates the effects that can be seen. Finally in Chapter 5 we conclude the report.

Chapter 2

The Ad hoc On-demand Distance Vector Protocol

AODV is a standardized routing protocol designed for mobile ad hoc networks [5]. The algorithm is on-demand. That is, routes between nodes are built when the source node needs them. AODV uses three routing packets to build a route to a destination: Route Request (RREQ), Route Reply (RREP) and Route Error (RERR). When a node does not have a route to a destination, which it want to communicate with, it broadcasts a Route Request in which it asks for a route to the destination.

When a node receives a Route Request, it sends a Route Reply, if the node is the destination node. An intermediate node can also respond to the request, if it has a fresh route to the specified destination in its route table. A route is considered fresh if the sequence number in the Route Request is lower than the corresponding value in the routing table or the sequence numbers are equal but the hop count is smaller.

The route is maintained as long as the route remains active. A route is considered active if data are sent from the source node to the destination node. If a link break occurs in an active route, a Route Error is propagated to the source node.

Furthermore, AODV uses sequence numbers to avoid old routes and the propagation of old information.

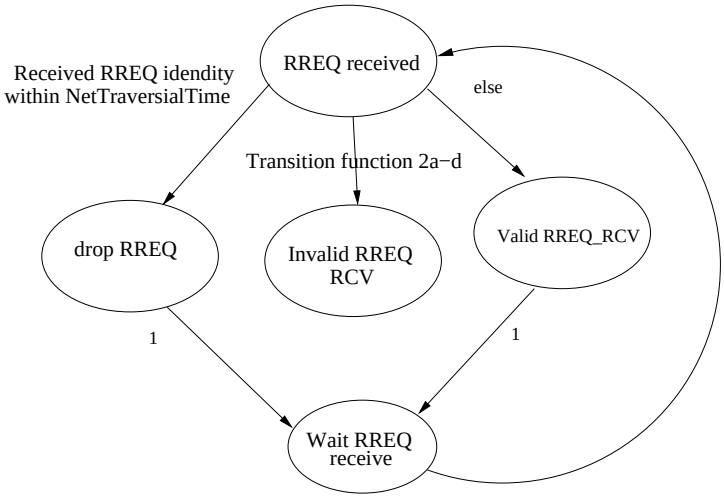

Figure 2.1: AODV Route Request received transition diagram.

2.1 The Extended Finite State Machine for AODV

In this section, we give an overview of the specification we have previously developed, a more detailed description can be found in [4].

The proposed AODV specification is an abstract of the AODV protocol specified in RFC 3561 [5]. That is, only essential details of the protocol are modeled in order to detect if any node performs attacks against other nodes. First, the network collection module of the IDS agent gathers routing packets received from neighbor nodes and routing packets sent to other nodes. Next, the AODV state machine examines the received routing packets from the node to detect if any node is performing routing attacks against other nodes. This is achieved by creating an instance of the AODV state machine for each received routing packet to a certain destination. If an instance of the AODV state machine already exists for the destination, the instance is updated with information from the received routing message.

AODV is modeled in two transition diagrams; Route Request and Route Reply. It is also possible to model RERR messages but we have not done that here. For each RREQ received from a unique destination, we create an instance of the AODV state machine that starts in the RREQ receive state, see figure

2.1. Similarly, we also create instances of the AODV state machine for every received RREP from a certain destination. Thus, there can be many instances of the state machine in runtime. It is important to find a way to limit the number of instances of the AODV state machine to save memory and computer capacity. Therefore, an instance of the AODV machine for a certain destination is deleted automatically when the state machine instance reach the final state (end-state). Thus, the maximal number of state machines is equal to the number of nodes in the network multiplied by two.

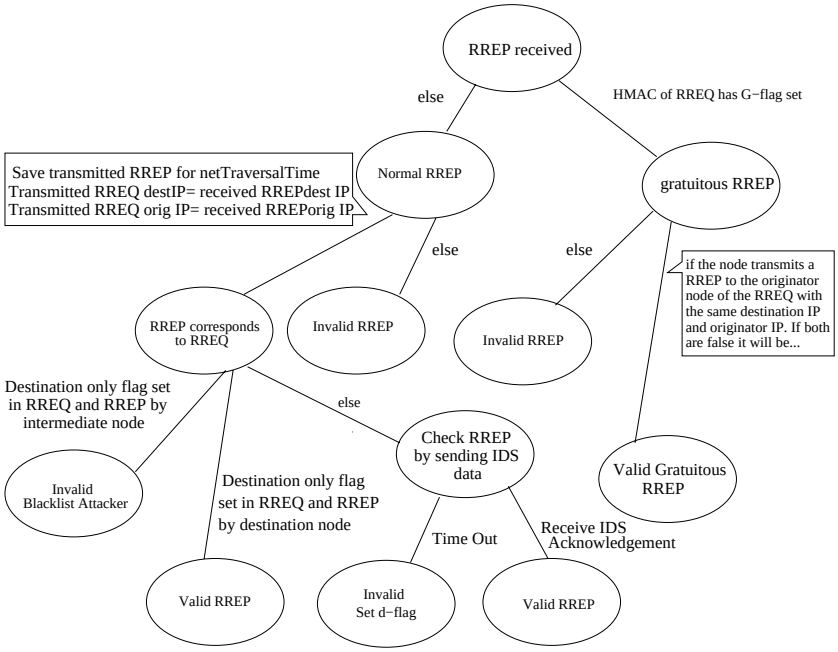

Figure 2.2: AODV Route Reply received transition diagram.

The RREQ transition diagram is depicted in figure 2.1 whereas the RREP transition diagram is depicted in 2.2. A more detailed description on how these specifications can be found in the previous report [4].

State machines of protocols often have more than one transition rule per state, i.e. a non-deterministic choice. In our proposal, there are several transition rules for some states, but they are mutually exclusive. That is, at any time only

one, or none, of the transitions is valid. Thus, a given input signal to a certain state always results in only one state.

For the Route Request transition diagram as shown in figure 2.1, the RREQ receive state has three transition functions with corresponding states;

1. Drop RREQ

   The RREQ is dropped if the node has received a RREQ with the same Originator IP address and RREQ identity within the last PATH_DISCOVERY_TIME.

2. Invalid RREQ

   A node broadcasting RREQ must follow certain rules. The state machine enters the invalid state if one of the following rules are true;

   (a) A neighbor node originates more than RREQ_RATELIMIT RREQ messages per second. This rule detects certain types of resource depletion attacks (attack 6 in table 1).

   (b) The waiting time between two RREQs with the same originator and destination IP address must be at least NET_TRAVERSAL_TIME. This rule detects certain types of resource depletion attacks which can for example be done by flooding the network with false RREQ messages.

   (c) A node identified by its signature has not incremented the RREQ identity when broadcasted a new RREQ. This rule detects for example rushing attacks, see [4].

   (d) Two nodes identified by their signatures originate a RREQ with the same originator IP address. This rule detects false message propagation attacks. Note that two nodes in a network never should use the same IP address. In case of dynamic IP addresses, a protocol such as DHCP (Dynamic Host Configuration Protocol) or DAD (Duplicate Address Detection) will make sure that two nodes do not use the same IP address. If static IP addresses are used, a certain signature should always correspond to a certain originator IP address.

3. Valid RREQ

If neither of the above translation function is true, the AODV state machine enters the valid RREQ state.

The Route Reply transition diagram As shown in figure 2.2, the RREP receive state has two transition functions with corresponding states;

1. Gratuitous RREP

   There are two types of RREPs; normal RREP and gratuitous RREP. If an originated node wants to have bidirectional communications with the destination node, it sets the gratuitous RREP flag (G-flag) in the RREQ. In such cases, any generation of a RREP by an intermediate node to the originated node should be accompanied by a gratuitous RREP to the destination node.

   We assume that an authentic gratuitous RREP is sent together with the received HMAC RREQ. Thereby, it is possible for another node to verify that the RREQ really had the G-flag set. Thus, it is possible to decide if the RREP is a gratuitous RREP or not.

2. Normal RREP

   If the RREP is not a gratuitous RREP it is considered as a normal RREP. The node first checks whether it has sent a corresponding RREQ, otherwise the RREP is invalid. For a valid RREP there are two cases. If the D-flag is set only the destination may respond, otherwise the RREP is invalid. However, if the D-flag is not set, the message needs to be further studied, see details in [4].

2.2 Weakness of AODV

It is possible to exploit a number of weaknesses in AODV to disrupt the communication between nodes. AODV does in the default implementation not support any way to verify that a node is who it claims to be, i.e. it is possible to lie about the originator IP address in a packet created. Another weakness is that a node can throw away packets unnoticed (black hole). If it also lies about having the shortest route (number of hops) to various nodes, more traffic will enter the hole. It is also possible to override legitimate packets by using alternate packets

with higher sequence numbers (rushing attack). See [3] for a complete threat analysis of the AODV protocol.

2.2.1 Specific Attacks to be studied

The attacks studied this time are more malicious than before, since they attack any node while they move around in the network. The earlier implementation only tried to disrupt communication between two specific nodes in a relatively fixed position network. Four attacks have been implemented from the original list of attacks, see [3]. The index number used is a reference to the attacks in Appendix B of [3].

2.2.2 Rush attack with RREQ [attack 2]

The purpose of this rush attack is to suppress a valid RREQ (request) sent by a real originator. When such a packet is received by the malicious node, it will broadcasts multiple RREQ messages with a range of RREQ_ID numbers impersonating the originator IP address, to the same destination. Future versions of the RREQ_ID number from the originator is not hard to guess since it increased by one at a time. A listener that hears the attack will prefer this route because of the higher RREQ_ID number, and discard any valid packets coming later with the same or lower RREQ_ID since it seems old.

2.2.3 False message propagation with RREQ [attack 3]

The goal of this attack is to reroute traffic through the malicious node, and then throw it away. It does this by broadcasting false RREQ packets, using all originators IP addresses to all destination IP addresses. When a node receives a false RREQ packet, it will update its routing table if the originator was not present or if the sequence number is higher. Next time the affected node wants to communicate with the originator, it will send packets to the malicious node. The attack is also known as a black hole, since the attacking node discards all traffic coming to it. The false packages are transmitted frequently to prevent future updates with valid routes.

2.2.4 False message propagation with RREP [attack 11]

In this attack, the malicious node reroutes traffic by using false RREP packets. Again, the purpose is to create a black hole and discard traffic. This is accomplished by unicasting RREP packets using all originators IP addresses and all destination IP addresses. The receiver stores the route if the originator address is new, the RREP sequence number is higher than stored value or if the number of hops to the destination is lower. The false packages are transmitted frequently to prevent future updates with valid routes.

2.2.5 False reply with RREP [attack 7]

This attack intercepts a request with an answer, hopefully before it reaches the final destination. An intermediate node (not the destination) replies on the RREQ sent, with a false RREP stating that it has a short route to the destination. Nodes receiving the false RREP will update their routing table to the malicious node, if the route seems the best way.

2.3 Attack/Defence

By introducing signatures [4] and sanity checks on the information in the packages, the nodes can successfully detect close to one hundred percent of the attacks. An overview of the defense state machine can be seen in Figures 2.1 and 2.2.

The new thing for this years implementation have been to incorporate the IDS in the simulator, to protect the nodes in real-time and prevent non-malicious nodes to spread the effects. A new response using blacklist have also been used for attack 7 since it can not be detected in real-time.

Chapter 3

Opimal Link State Routing

Optimal Link State Routing, OLSR, is one of the few ad hoc-routing protocols that have reached the status of RFC, and besides AODV it is the routing protocol that have been shown most interest in the research community. It is published as RFC 3626 [7]. Unlike AODV which only finds routes when they are needed OLSR is a proactive protocol, this means that the routing protocol will attempt to build routes between all nodes independent on whether they are needed or not. This has the advantage that a path already exists when it is needed and no route search must be done before user traffic can be sent. The disadvantage is in some scenarios more overhead traffic. OLSR is most beneficial if many nodes often want to communicate in the network so that information about most paths will be needed (thereby giving little unnecessary overhead). This is often the case in military networks since multicast to all other nodes is a common type of traffic.

In this chapter we will first give a short description of OLSR (for more details we refer to RFC 3626 [7]). Then we will show that the same principals that we used for AODV can be used to develop a similar specification-based IDS for OLSR.

3.1 Overview of OLSR

OLSR is based on the classic Link State Routing (LSR) protocol but with some changes that make it more useful for mobile ad hoc networks with low link ca-

pacity. In LSR each node sends information about all links to the entire network, thereby making it possible for each node to calculate the route with least cost for each destination. However, this protocol generates very high overhead traffic which makes it impractical for mobile networks, where changes in topology is common.

In OLSR the overhead information is decreased by letting each node choose a subset of neighbors called multipoint relays (MPR) that are the only nodes that will retransmit a message. These MPRs are chosen so that all two-hop neighbors will be reached if all MPRs retransmit the control messages. This reduces the number of needed retransmission of the link state messages (especially for a dense network). In addition, all links to these neighbors needs to be symmetric, that is communication in both directions, must be possible.

The second method to decrease routing overhead compared to LSR is that OLSR only sends partial link state information rather than sending information of all links. The minimum required information is that all nodes chosen to be MPR nodes send information about the links to those nodes that selected them as MPR, although more information than this may be sent for redundancy purpose.

The control messages used are sent periodically, therefore a reasonable message loss can be accepted by the protocol without significant degradation.

3.1.1 MPR selection

Each node must select, among its one hop neighbors, a subset that will be MPRs. However in the OLSR specification only a suggested heuristic algorithm is proposed and there is no hard requirement that it must be used. In short, the algorithm starts by looking for two-hop neighbors that can only be reached through a single neighbor, then setting these neighbors to be MPRs.

One important feature is the “willingness” for a node to become a MPR. This is set by each node and goes from “will never” to “will always”. Nodes advertising “will always” must be chosen as MPRs. In the next step the algorithm looks on those neighbors with highest “willingness” and assigns those in the order of the highest number of two-hop neighbors they can reach. If not all two-hop neighbors are reached when this group have been added, groups with lower “willingness” will be tested.

Information about MPR selection is included in the HELLO messages.

3.1.2 Packet Formats, Forwarding, and Processing Messages

OLSR communicates using a unified packet format for all data related to the protocol.

Upon receiving a basic packet, a node examines each of the message headers. Based on the value of the Message Type field, the node can determine how to handle each message. A node may receive the same message several times. Thus, to avoid re-processing of some messages which were already received and processed, each node maintains a Duplicate Set. In this set, the node records information about the most recently received messages in order to avoid duplicate processing of a message.

3.1.3 OLSR Control Traffic

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

HELLO messages are exchanged periodically among neighbor nodes, in order to detect links to neighbors, to detect the identity of neighbors and to signal MPR selection. Upon receiving a HELLO message, a node examines the lists of addresses. If its own address is included, it is confirmed that bi-directional communication is possible between the originator and the recipient of the HELLO message. When a link is confirmed as bi-directional, this is advertised periodically by a node with a corresponding link status of “symmetric”. In addition, to information about neighbor nodes, periodic exchange of HELLO messages allows each node to maintain information describing the links between neighbor nodes and nodes which are two hops away. This information is recorded in a nodes 2-hop neighbor set and is explicitly utilized for the MPR optimization.

TC messages are periodically flooded to the entire network, in order to spread link state (topological) information to all nodes. A TC message contains a set of bi-directional links between a node and a subset of its neighbors. The topological information is used in the MPR optimization. Only nodes which have been selected as an MPR generate (and relay) TC messages. The TC message contains an ANSN field which contains the Advertised Neighbor Sequence Number. This number is associated with the node’s advertised neighbor set, and is incremented each time the node detects a change in this set.

There are two more types of control messages in OLSR: MID (Multiple

Interface Declaration) and HNA (Host and Network Association). MID messages are only generated by nodes with multiple OLSR interfaces, in order to announce information about its interface configuration to the network. HNA messages are only generated by nodes with multiple non-OLSR interfaces, and have the purpose of providing connectivity from an OLSR network to a non-OLSR network.

3.2 Possible attacks

In this section, we will very shortly describe some possible attacks on OLSR. For more detailed description and more attacks see [3], [18] and [19].

In OLSR each node has two different responsibilities. Firstly, each node must correctly generate routing protocol control traffic according to the protocol specification. Secondly, each node must forward control traffic generated in other nodes in the network. Hence, incorrect behavior of a node can result from either a node generating incorrect control messages or from incorrect relaying of control traffic from other nodes.

3.2.1 Incorrect traffic generation

A node can misbehave by generating false HELLO, TC or MID/HNA messages. We can observe that this can be done in two different ways: through generating control traffic pretending to be another node or through transmitting incorrect information in control messages.

To exemplify incorrect traffic generation we look at HELLO messages. A misbehaving node, E, may send HELLO messages pretending to be another node, C (see Figure 3.1). This will result in nodes A and B announcing that C is a one hop neighbor in their HELLO and TC messages. Conflicting routes to node C with possible loops or connectivity loss may result from this.

A misbehaving node may also send HELLO messages containing incorrect information about its set of neighbors. This can be done in two ways: sending out an incomplete set of neighbors or stating non-neighbors are neighbors. In the first case the network may be without connectivity to the ignored neighbors. In the second case nodes may select wrong set of neighbors as MPRs with the result that some nodes may not be reachable in the network.

3.2.2 Incorrect traffic relaying

If a node do not relay control messages in a proper way, network malfunctions are possible. For example if a node do not relay TC messages, the network may experience connectivity problems. In networks where no redundant path exists connectivity loss will be the result, but other topologies may provide redundant connectivity and routes can still be found.

If MID and HNA messages are not properly relayed, information about multiple nodes interfaces and connection to other networks may be lost.

Another attack is replaying old control messages. This causes nodes to record out of date topology information. However, a control message ca not be replayed as is since nodes already received it will ignore the replayed message because of the MSN, Message Sequence Number (and ANSN for TC messages). The attacker needs to increase MSN (and ANSN for TC) to cause messages losses or connectivity problems.

A misbehaving node may also not forwarding data packets. Hence, data packet transmitted along routes containing the misbehaved node will not reach their destination.

3.3 Specification-based IDS for OLSR

Since some work on specification-based IDS for OLSR already exists, one paper was written by J-M Orset et al [20] which gives a rather good description on how such a method can be used for OLSR, we will not develop an IDS for OLSR on

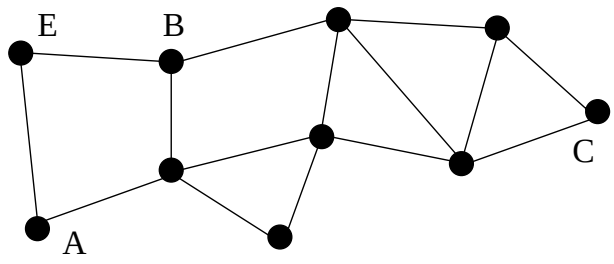

Figure 3.1: Node E sends HELLO messages pretending to be node C.

our own, but rather describe their solution and give some comments on how it can be expanded.

The IDS developed in [20] has some limitations though. They assume that each node only has one interface (OLSR is specified to be able to handle several) and only has one link to each node. These assumptions are similar to what we have earlier used.

Further, although there are two types of messages in OLSR, HELLO messages between neighbors and TC messages that updates further away, only HELLO messages are being modelled with an EFSM. Modelling also TC messages would be necessary in a real scenario.

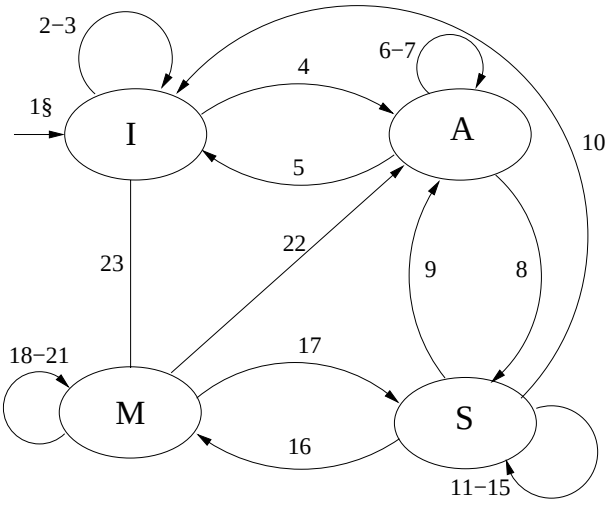

Figure 3.2: OLSR HELLO message transition diagram.

In figure 3.2 we show a picture of their OLSR EFSM for each link a node sets up to its neighbors. Notice, that there is one EFSM run independently for each of a nodes links. In this EFSM there are four states (I, A, M, and S).

The first state (I) is the initial state, in which case the node has not heard any neighbor, and send neighbor discovery packets according to its timers.

A node enters the second state (A - asymmetrical) if it hears a HELLO message from another node. It cannot yet claim a symmetric link to the other node. From this state the node can return to the first state if it receives no more messages or change into the third step (S - symmetric) at which case the link

can be used by OLSR.

However, this transition will only be allowed with correct information in the received HELLO packet. If the other node claims an asymmetric link to the present node the transition from A to S can only be allowed if the present node sent an empty HELLO (more specifically one without information about the other node). If the other node claims a symmetrical link such a message will only be considered correct if we are not still in the initialization state.

Once the node reach the symmetric state the nodes can choose each other as MPR nodes. If the node starts to hear empty HELLO messages (regarding its existence) it means that the neighbor cannot hear it anymore and it will return to asymmetric state (since it can still hear its neighbor).

When a node is chosen as an MPR for one of its neighbors, it now have responsibility to forward data from its selectors. It is important that the data really comes from the selector. In this state the node must also generate link state messages. As long as the other node keeps it selected as a MPR the node will stay in this state. Depending on messages it can change back to any of the previous messages.

A more detailed description of all transitions in figure 3.2 is given in [20].

Now, this model can be used to detect possible attacks or implementation errors that causes events that is not acceptable according to the EFSM. In its present form it cannot directly be used versus all attacks, especially the dealing with TC messages are lacking. Further, there are no included protection with help of cryptography which we have assumed with AODV. This means that a node can usurp another's identity as long as communication between nodes is handled as it should. However, such protection can be added for OLSR in the same way as for AODV so it could be expanded in order to prevent such attacks as well.

3.4 Some Concluding Remarks about OLSR

AODV and OLSR represents the two different types of routing principles, reactive and proactive. Although the specification based IDS we have described here for OLSR is not complete, it suggests that there will be no problems in completing it.

In many situations OLSR is probably more relevant for military scenarios

than AODV is. Situation awareness is a typical services where all nodes need to send information to all other nodes and a proactive protocol is usually better at handling such traffic.

Expanding the IDS for OLSR so that protecting that kind of service against hostile nodes would be a very interesting continuation of the OLSR work.

Chapter 4

Evaluation

In this chapter we will describe the results from the evaluation of the simulated attacks on AODV. The specific attacks we have simulated was described in chapter 2. We start with a description of the used scenario.

4.1 Evaluated Scenarios

In our scenario, we have 32 nodes moving randomly in an area of 4x4 km. They move independently of each other at a constant speed of 2 m/s. The nodes randomly change direction at certain intervals. When a node reaches the area border, it turns and proceeds in a new direction. The scenario is running during 3600 seconds. Due to the mobility, the amount of node pairs that have single- or multi-hop connections will vary. In the generated networks, approximately 95% of all pairs of nodes were connected. The same node movements were used for all simulations.

The traffic in the network is modeled as a number of file transfer sessions with specified application packet sizes and end-to-end delay requirements. A session is considered successful if each of the received packets is delayed less than 2 s. We vary the average total number of running sessions and evaluate their success rate with respect to the delay requirements. The traffic in each session is modeled as point-to-point traffic from a source node to a destination node. New sessions start according to a Poisson process. Furthermore, we assume that the traffic is uniformly distributed over the nodes, i.e. each node is equally probable

as the source node and each node (except the source node) is equally probable as the destination node. The average length of a file transfer session is 12 seconds, and during each session, the source transmits packets to the destination with a constant bit rate of 12.2 kbps.

An essential part of modeling an on-ground or near-ground radio network is the electromagnetic propagation characteristics due to the terrain variation. A common approach is to use the basic path-loss, L_b, between two nodes (radio units). To estimate the basic path-loss between the nodes, we use a Uniform geometrical Theory of Diffraction (UTD) model by Holm [21]. To model the terrain profile, we use a digital terrain database. All our calculations of the basic path-loss are carried out using the wave propagation library DetVag-90® [22].

For a sending and a transmitting node with isotropic antennas (antenna gain equal to one), we define the signal-to-noise ratio (SNR) in the receiving node as follows:

$$\frac{P}{NL_b R}, \tag{4.1}$$

where P denotes the power of the transmitting node (equal for all nodes), N is the receiver noise power, R is the data rate, and L_b is the basic path-loss between the nodes. In the simulations, we assume that the radio system has a fixed link data rate $R = 1\,\mathrm{Mbps}$ in absence of interference from other nodes.

On the multiple access level, we use TDMA (Time Division Multiple Access), a static collision-free protocol where the channel sharing is carried out in the time domain. This means that the time is divided into time slots and each node is assigned one or several time slots where it is allowed to use the channel. We want to use a schedule that adapts to the traffic and the network topology. Therefore, we use an optimal method (with respect to resource utilization) to decide which node may use a certain slot and do not consider the required control traffic. According to this method, we determine at the beginning of each time slot, which node has the oldest queued packet. This node is then allowed to use the time slot. The protocol is thus traffic adaptive, i.e. the node is allocated time slots corresponding to the traffic load the node is exposed to. For simplicity, the slot assignment in our simulation is centralized.

In the simulations, we pick one of the nodes in the scenario to be an attacking node. We assume that the authentication in the intrusion detection makes the RREP and RREQ packets in AODV 128 bits longer.

4.2 Implementation of AODV

Our AODV implementation tries to follow the specification in RFC 3561 [5], as much as possible, however, the specification is unclear at some points. Also, practical implementations will need to add additional tests that prevents an application from crashing due to errors in received packets, since how to handle all such incorrect messages are not specified in the RFC.

Due to this, we have made some additional assumptions when implementing the algorithm to avoid problems with the simulator. For simplicity, we have assumed that there is a link layer protocol that finds the direct link neighbors. Only information from this protocol have been used to set up routes to neighbors. This means that a received RREQ with a false identity will not necessarily cause a node to set up an incorrect route to that node if it already is a neighbor.

There is of course nothing that directly prevents an evil node to falsify such neighbor information as well (by attacking the neighbor discovery protocol), but this is not a network layer protocol, so we have not implemented such an attack. Furthermore, the same protection that will be used in the IDS to prevent false identities can of course be used by such messages as well. This have the consequence that disrupting routes between neighboring nodes are difficult for a malicious node.

4.3 Results

In this section we show the results from our simulations of the 32-node mobile network. In [4] we presented the results for a static 8-node network without actually simulating the additional overhead caused by our IDS. In that case detection rate of attacks where close to 100% with only some few cases of false alarms.

However, static networks make the problem of detection considerably simpler. In table 4.1 we show the results for a mobile network with 32 nodes and 5 simultaneous sessions during a 300 seconds long simulation run (as an example, for the rest of the results we are using a 3600 seconds long simulation).

As can bee seen the results are still very good, we are close to having a 100 % detection rate with very few cases of false alarms. However, we can see that the IDS do miss to detect attacks in some few cases and in some rare cases

Description	RREQ false message propagation	RREP false message propagation	RREP false reply	RREQ rushing
Packets	503771	159317	98427	132102
Bad Packets(%)	81,55	40,52	6,3	14,1
Detection rate (%)	100	99,96	98,16	100
False alarms (%)	0,0018	0,011	0,0031	0
False Blacklisting of nodes	4	14	12	0

Table 4.1: Detection rates of the different attacks.

even blacklist nodes. It is mainly for attacks using RREPs that we can see this, since RREPs can be answered by intermediate nodes and thereby not as easily be protected by signatures.

It is also interesting to note that even when a node gets blacklisted by another node it does not necessarily prevent it from communication with other nodes as long as there are paths through other nodes that have not blacklisted it.

Due to the implementation of AODV (nodes believe more in information from the neighbor protocol than AODV) and that we do not have any attacks versus the neighbor protocol none of the attacks will have any effect on two nodes communicating within one-hop range.

Another important issue is the additional overhead caused by the IDS. In the following we will will study the success-rate of sessions with and without attacks and IDS to see how much our IDS costs in terms of success rate of sessions.

4.3.1 Rushing attack - attack 2

In figure 4.1, simulation results from the rushing attack are shown. We plot the percentage of successful sessions as a function of the average number of simultaneous sessions run, a session is considered successful if 95 % of the packets arrive within the prescribed time limit.

Without the IDS the attack is highly successful, however, the effect we see can probably in part be seen as a denial of service attack since in this attack an attacker must generate a large amount of packets to prevent the nodes from communicating which will consume a lot of resources when the are broadcasted

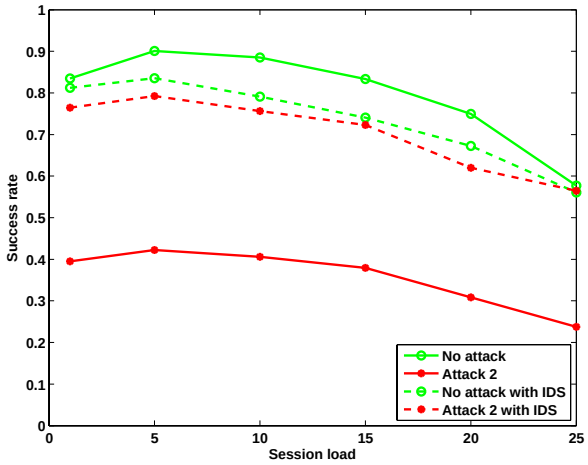

Figure 4.1: Average ratio of successful sessions for different amount of average sessions run of attack 2.

over the network.

The IDS prevents such spreading though and minimize the effect of the attack, which can be noticed since the percentage of bad packets of the attacks is not that high anymore, see table 4.1,

4.3.2 False RREQ - attack 3

In figure 4.2, simulation results from attack 3 are shown. This is a form of a black hole attack. The attacker uses RREQ messages to try to force the other nodes to use it as a relay. As can be seen in Figure 4.2 it is a very efficient attack versus the network, however, to block all communication (which we have tried with all attacks) this attack needs to generate very many packets as can be seen in table 4.1, which means that it also will have a clear denial-of-service effect.

Further, the attack gets much more efficient if we use a higher sequence number and RREQ ID, since this essentially also makes it a rushing attack. Using too low values makes the other nodes ignore the messages which gives very little effect on performance.

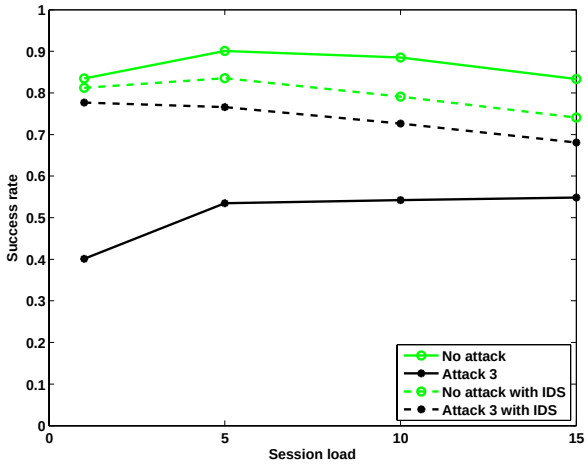

Figure 4.2: Average ratio of successful sessions for different amount of average sessions run of attack 3.

This attack is probably more useful if an attacker wants to target the communication of single nodes.

4.3.3 False reply with RREP - attack 7

In figure 4.3, we show results the case from when an attacker replies to genuine RREQ messages with a false RREP. In this case the attacker attempts to create a black hole effect. The attack works very well for a small number of sessions in the network, but less well if the average number of sessions increase.

If the network have many sessions running this also means that there is much more routing information in the network. More nodes can faster reply to a RREQ with correct informations so the attack will be less efficient for large traffic loads which we can see.

Also for this attack the IDS works very well.

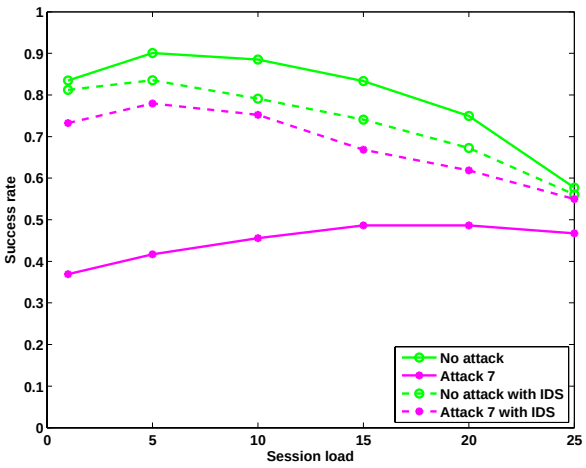

Figure 4.3: Average ratio of successful sessions for different amount of average sessions run of attack 7.

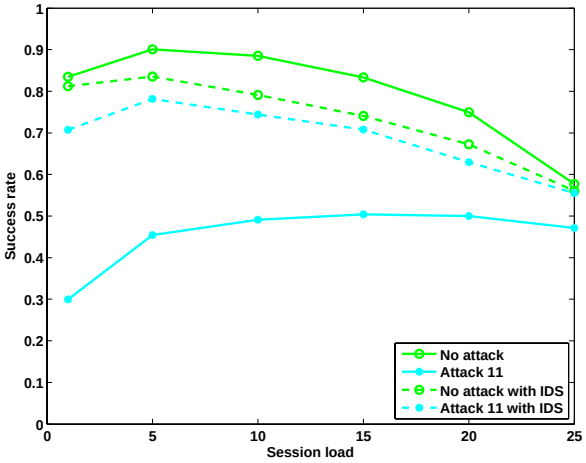

Figure 4.4: Average ratio of successful sessions for different amount of average sessions run of attack 11.

4.3.4 False message propagation with RREP - attack 11

In figure 4.4 we plot the results from the case when an attacker sends RREP into the network without any RREQ to trigger them.

For a lowly loaded network, this is one of the most efficient attacks but as the network load increases we also here see a slight increase in success rate of the network sessions.

Also in this case does the IDS mostly remove the effect of the attack.

4.4 Concluding Remarks

Most attacks can be very efficient in disrupting an ad hoc network.

Here attacks versus communication between neighbors was not possible, but in general this is not true, we did just not implement any such attack.

Our IDS works very well, almost no missed attacks and even the denial of service effect of the attacks seems highly limited.

Chapter 5

Conclusions

In this report we have expanded our research on vulnerabilities of mobile networks from [4] by studying more realistic networks.

We have shown by simulations that some of the well known attacks on AODV do have a large reduction of the success rate of communication sessions, preventing more or less all nodes from communicating.

However, our specification-based IDS removes almost all of the effects of the attacks with very little cost in terms of overhead and false alarms. In addition, using the IDS on networks that are not attacked will only slightly decrease the performance of the network.

We have also studied OLSR and show that the same methods could be applied also on this protocol.

5.1 Future Work

OLSR have several properties that is interesting for military networks, especially for handling broadcast traffic. Continued work to develop the IDS for OLSR would therefore be important, since broadcast traffic have little priority in civilian research.

Bibliography

[1] Y. Zhang, W. Lee, and Y. Huang, "Intrusion detection techniques for mobile wireless networks," *Mobile Computing and Communications Review*, vol. 7, no. 1, pp. 74–94, jan 2003.

[2] S. Axelsson, "Intrusion detection systems: A taxonomy and survey," Tech. Rep. Tech. Report no. 99-15, Dept. of Comp. Eng., Chalmers Univ. of Technology, 2003.

[3] E. Hansson, J. Grönkvist, and J. Nilsson, "Intrångsdetektering i mobila ad hoc-nät," Technical Report FOA-R--1375--SE, Swedish Defence Research Agency., Div. of Command and Control. Linköping, Sweden, 2005, (In Swedish).

[4] E. Hansson, J. Grönkvist, K. Persson, and D. Nordqvist, "Specification-based intrusion detection combined with cryptography methods for mobile ad hoc networks," Technical Report FOA-R--1867--SE, Swedish Defence Research Agency., Div. of Command and Control. Linköping, Sweden, 2005.

[5] C. Perkins et al, "On-demand distance vector (AODV) routing," *RFC 3561*, 2003.

[6] D. Johnson, D. Maltz, and Y-C. Hu, "The dynamic source routing protocol for mobile ad hoc networks (DSR)," *draft-ietf-manet-dsr, internet draft*, (work in progress).

[7] T. Clausen and P. Jacquet, "Optimised link state routing protocol (OLSR)," *RFC 3626*, 2003.

[8] R. Ogier et al, "Topology dissemination based on reverse-path forwarding (TBRPF)," *RFC 3684,*, 2004.

[9] Y. Hu and A. Perrig et al., "Ariadne: A secure on-demand routing protocol for ad hoc networks," in *Proceedings of MOBICOM 2000*, 2002, pp. 275–283.

[10] M. Guerrero, "SAODV," *draft-guerrero-manet-saodv, internet draft*, (work in progress).

[11] R. Ramanujan et al, "Techniques for intrusion-resistant ad hoc routing algorithms (TIARA)," in *Proc. of MILCOM 2000*, oct 2000, vol. 2, pp. 660–664.

[12] Y. Zhang and W. Lee, "Intrusion detection in wireless ad hoc networks," in *Proceedings of MOBICOM 2000*, 2000, pp. 275–283.

[13] S. Bhargava and D.P. Agrawal, "Security enhancements in AODV protocol for wireless ad hoc," in *Proceedings of VTC Fall 2001*, oct 2001, vol. 4, pp. 2143–47.

[14] Marti et al., "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. 6th Annual int Conf. Mobile Comp. and Net.*, 2001, pp. 255–65.

[15] A. B. Smith, "Examination of an intrusion detection architecture for wireless ad hoc networks," in *Proceedings of 5th Nat. Colloq. For Info Sys. Sec. Education*, may 2001.

[16] P. Albers et al., "Security in ad hoc networks: a general intrusion detection architecture enhancing trust based approaches," in *In Proc. 1st Int. Wksp. Wireless Info. Sys.*, apr 2002.

[17] C-Y Tseng et al., "A specification-based intrusion detection system for AODV," in *In Proc. Of ACM Workshop on Security of ad hoc and sensor networks*, 2003.

[18] Adjih et al, "Securing the OLSR protocol," in *Proc. of MedHoc 2003*, june 2003, vol. 2.

[19] D. Raffo, "Security schemes for the OLSR protocol for ad hoc networks," Doctoral thesis, INRIA Rocquencourt, sep 2005.

[20] J-M. Orset, B. Alcalde, and A. Cavelli, "An EFSM-based intrusion detection system for ad hoc networks," in *In proc. ATVA 05*, 2005.

[21] P. D. Holm, "UTD-diffraction coefficients for higher order wedge diffracted fields," *IEEE Trans. Antennas Propagat.*, vol. AP-44, no. 6, pp. 879–888, June 1996.

[22] B. Asp, G. Eriksson, and P. Holm, "Detvag-90® — Final Report," Vetenskaplig Rapport FOA-R–97-00566-504–SE, Försvarets Forskningsanstalt, Avdelningen för ledningssystemteknik, Linköping, 1997.