A Specification-based Intrusion Detection System for AODV

Chin-Yang Tseng[1] Poornima Balasubramanyam[1] Calvin Ko[2]
Rattapon Limprasittiporn[1] Jeff Rowe[1] Karl Levitt[1]

[1]Computer Security Laboratory
University of California, Davis
{ctseng, pbala, rlim, rowe, levitt} @ucdavis.edu

[2]Network Associates Laboratories
Network Associates, Inc.
calvin_ko@nai.com

ABSTRACT
The Ad hoc On-Demand Distance Vector (AODV) routing protocol, designed for mobile ad hoc networks, offers quick adaptation to dynamic link conditions, low processing and memory overhead, and low network utilization. However, without keeping in mind the security issues in the protocol design, AODV is vulnerable to various kinds of attacks. This paper analyzes some of the vulnerabilities, specifically discussing attacks against AODV that manipulate the routing messages. We propose a solution based on specification-based intrusion detection to detect attacks on AODV. Briefly, our approach involves the use of finite state machines for specifying correct AODV routing behavior and distributed network monitors for detecting run-time violation of the specifications. In addition, one additional field in the protocol message is proposed to enable the monitoring. We illustrate that our algorithm, which employs a tree data structure, can effectively detect most of the serious attacks in real time and with minimum overhead.

Keywords
AODV, MANET, intrusion detection, specification-based detection, network monitor, P2P network.

1. INTRODUCTION
A mobile ad hoc network (MANET) is a collection of mobile computers or devices that cooperatively communicate with each other without any pre-established infrastructures such as a centralized access point. Computing nodes (usually wireless) in an ad hoc network act as routers to deliver messages between nodes that are not within their wireless communication range. Because of this unique capability, mobile ad hoc networks are envisioned in many critical applications (e.g., in battlefields). Therefore, these critical ad hoc networks should be sufficiently protected to achieve confidentiality, integrity, and availability.

The dynamic and cooperative nature of MANETs presents substantial challenges in securing these networks. Unlike wired networks which have a higher level of security for gateways and routers, ad hoc networks have the characteristics such as dynamically changing topology, weak physical protection of nodes, the absence of centralized administration, and highly dependence on inherent node cooperation. As the topology keeping changing, these networks do not have a well-defined boundary, and thus, network-based access control mechanisms such as firewalls are not directly applicable. In addition, there is no centralized administration, making bootstrapping of crypto systems very difficult. It is extremely easy for a malicious node to bring down the whole network. As a result, ad hoc networks are vulnerable to various attacks including eavesdropping, spoofing, modification of packets and distributed denial-of-service (DDoS) attacks.

Security services, such as authentication services and access controls, can enhance the security of ad hoc networks. Nevertheless, these preventive mechanisms alone cannot deter all possible attacks (e.g., insider attackers possessing the key). Therefore, it is necessary to have other security mechanisms to deal with misbehaving insider nodes that possess the valid key and access rights. Intrusion detection, which has been successfully used in wired networks to identify attacks, can provide a second line of defense. In particular, intrusion detection and response capability is very important as many of the real ad hoc networks will be deployed in hostile environments in which legitimate nodes could be captured and used by adversaries.

Intrusion detection involves the runtime gathering of data from system operation, and the subsequent analysis of the data; the data can be audit logs generated by an operating system or packets "sniffed" from a network. Intrusion detection techniques can be mapped into three concepts: signature-based detection, anomaly detection, and specification-based detection. In signature-based intrusion detection [5][10], the data is matched against known

attack characteristics, thus limiting the technique largely to known attacks, even excluding variants of known attacks.

In anomaly detection [6], profiles of normal behavior of systems, usually established through automated training, are compared with the actual activity of the system to flag any significant deviation. A training phase in anomaly-based intrusion detection determines characteristics of normal activity; in operation, unknown activity, which is usually statistically significantly different from what was determined to be normal, is flagged as suspicious. Anomaly detection can detect unknown attacks, but often at the price of a high false alarm rate.

In specification-based detection [7][8], the correct behaviors of critical objects are manually abstracted and crafted as security specifications, which are compared with the actual behavior of the objects. Intrusions, which usually cause object to behavior in an incorrect manner, can be detected without exact knowledge about them. So far, specification-based detection has been applied to privileged programs, applications, and several network protocols.

This paper describes our on-going research on intrusion detection for mobile ad hoc networks. In particular, we employ specification-based techniques to monitor the ad hoc on-demand distance vector (AODV) routing protocol, a widely adopted ad hoc routing protocol. AODV is a reactive and stateless routing protocol that establishes routes only as desired by the source node. AODV is vulnerable to various kinds of attacks [1]. This paper analyzes some of the vulnerabilities, specifically discussing attacks against AODV that manipulate the routing messages. We propose a solution based on the specification-based intrusion detection technique to detect attacks on AODV. Briefly, our approach involves the use of finite state machines for specifying correct AODV routing behavior and distributed network monitors for detecting run-time violation of the specifications. In addition, one additional field in the protocol message is proposed to enable the monitoring. We illustrate that our algorithm, which employs a tree data structure and a node coloring scheme, can effectively detect most of the serious attacks in real time and with minimum overhead.

This research is the first effort to apply specification-based detection technique to detect attacks in ad hoc network that manipulate routing messages to achieve the attack goal. In particular, we present the specification of AODV that describes the valid flow of AODV routing messages. In addition, distributed network monitors are used to monitor whether the nodes conform to the specification.

The remainder of this paper is organized as follows: Section 2 describes relative work on AODV security and intrusion detection on ad hoc networks. Section 3 gives an overview of vulnerabilities in AODV and present several example attacks to motivate the research. Section 4 describes the specification-based approach to monitoring AODV. In addition, it describes the specifications, the monitoring algorithm, and the architecture of the specification-based IDS. In Section 5, we illustrate how our approach and detect various kinds of attacks mentioned in section 2. Finally, we discuss future work in section 6 and summarize the paper in section 7.

2. RELATED WORK

The related work roughly may be categorized into two sub-areas: authentication-based approaches directed at the ad-hoc routing protocols, and general IDS targeted at mobile ad hoc networks. The primary work in securing routing protocols has addressed the problem of implementing effective integrity mechanisms. Approaches that impose authentication and integrity mechanisms are found in ARAN [4][15] and S-AODV[17] among others. The latter also includes a useful description of exploits in the ad hoc domain.

2.1 IDS Approaches for Mobile Ad-hoc Networks

One of the first proposed approaches for an integrated IDS architecture is in [19]. They present a cooperative distributed intrusion detection and response framework for MANET. Anomaly detection is the primary ID approach discussed, including anomalies in routing updates, abnormalities at the MAC layer (number of channel requests, etc.) and at the mobile application layer (number of requests to a service, duration of service requests etc.). Ramanujan et. al. present a system to detect, avoid, and recover from malicious attacks. They introduce three key ideas – a distributed firewall mechanism to limit the impact of flooding, an algorithm to detect and recover from intruder induced path failures, and a wireless router extension architecture which allows these techniques to be incorporated into existing wireless IP routers. Kachirski and Guha [9] describe a wireless IDS for ad hoc networks based on mobile agent technology. The system uses agents at various levels and aggregates their results at some cluster points that are elected using distributed algorithms. The idea is to distribute the IDS functionality between the nodes to minimize the total IDS-related processing time by each node.

Rao and Kesidis [13] propose a statistics based approach. The idea is to estimate the congestion at intermediate nodes and decide if the intermediate node is not forwarding packets at the desired rate because of congestion or because of malicious behavior. The work described in [2],[3],[11], and [12] use the mechanism of assigning a value to the "reputation" of a node and using this information to weed out misbehaving nodes and use only trusted and verifiably good nodes. Primarily, the intrusive activity addressed is that of misbehaving nodes that agree to forward packets to neighbors, but fail to do so. Passive eavesdropping is employed in monitoring the nodes in the first three approaches. This monitoring choice suits the nature of the domain where nodes can eavesdrop over other nodes within radio range and use that to isolate malicious nodes. In both [11] and [12], the authors implement their IDS approach on top of the DSR protocol. Belding-Royer [20] employ an IDS approach that is based on a stateful analysis of the data of AODV control packet streams in order to detect intrusions. This approach is based on the State Transition Analysis Technique (STAT) developed initially to model host and network based intrusions in a wired environment. In the current implementation, a sensor is deployed individually in each of a subset of nodes, and the sensors do not communicate with each other.

2.2 Authentication Approaches

In [15], a new cryptographic protocol, ARAN, is proposed. They assume that every node has its own public and private keys distributed by a trusted sever. The originator sends out a RREQ with its signature, and each intermediate node will verify the signatures of the previous intermediate node and the sender, and sign the packet sent by the originator. (The signature of previous intermediate node is discarded) Zapata and Asokan [18] propose S-AODV, which shares the same approach. Both of them use signatures to protect the AODV header from being modified and keep the header readable. However, insider attacks such as the tunneling attack shown in section 3.4, still remain a problem.

3. VULNERABILITIES IN AODV

ADOV is vulnerable to many different types of attacks [1]. In this section, we examine specific vulnerabilities in AODV that allow subversion of routes. In addition, we provide several attack scenarios that exploit the vulnerabilities to motivate our research.

3.1 Overview of AODV

The Ad hoc On-demand Distance Vector (AODV) routing protocol is a reactive and stateless protocol that establishes routes only as desired by a source node using route request (RREQ) and route reply (RREP) messages. When a node wants to find a route to a destination node, it broadcasts a Route Request (RREQ) message with a unique RREQ ID (RID) to all its neighbors. When a node receives a RREQ message, it updates the sequence number of source node and sets up reverse routes to the source node in the routing tables. If the node is the destination or the node has a route to the destination that meet the freshness requirements1, it unicasts a route reply (RREP) back to the source node.

The source node or the intermediate nodes that receives RREP will update its forward route to destination in the routing tables. Otherwise, it continues broadcasting the RREQ. If a node receives a RREQ message that has already processed, it discards the RREQ and does not forward it.

In AODV, sequence number (SN) plays a role to indicate the freshness of the routing information and guarantee loop-free routes. Sequence number is increased under only two conditions: when the source node initiates RREQ and when the destination node replies with RREP. Sequence number can be updated only by the source or destination. Hop count (HC) is used to determine the shortest path and it is increased by 1 if RREQ or RREP is forwarded each hop. When a link is broken, route error packets (RERR) are propagated to the source node along the reverse route and all intermediate nodes will erase the entry in their routing tables. AODV maintains the connectivity of neighbor nodes by sending hello message periodically.

[1] A route is considered to be fresh enough if the corresponding sequence number is greater than that contained in the RREQ; or equal to that contained in the RREQ and meanwhile hop count is smaller than that contained in the RREQ [1].

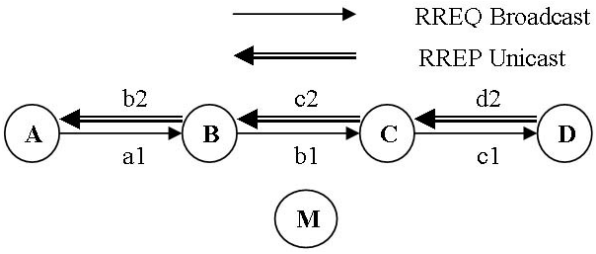


Figure 1: An AODV Scenario

Figure 1 illustrates the flow of the RREQ and RREP messages in a scenario wherein a node A wants to find a route to a node D. (Initially, nodes A, B, C and D do not have routes to each other). A broadcasts a RREQ message (a1), which reaches B. B then re-broadcast the request (b1). C recieves the messages and broadcasts the message (c1), which arrives at the destination node D. Last, D unicasts back the RREP message to A. We call these RREQ and RREP packets a request-reply flow. The values of the fields in the routing messages are denoted in Table 1.

Table 1: Values of RREQ and RREP

Type	RREQ			RREP		
Msg	a1	b1	c1	d2	c2	b2
IP.Src	A	B	C	D	C	B
IP.Dst	255	255	255	C	B	A
HC	0	1	2	0	1	2
AODV.Dst	D			D		
SN.Dst	0 (Unknown)			61		
AODV.Src	A			A		
SN.Src	100					
RREQ ID	20					

3.2 Vulnerable Fields in AODV Control Messages

In general, AODV is efficient and scalable in terms of network performance, but it allows attackers to easily advertise falsified route information to redirect routes and to launch various kinds of attacks. In each AODV routing packet, some critical fields such as hop count, sequence numbers of source and destination, IP headers as well as IP addresses of AODV source and destination, and RREQ ID, are essential to the correct protocol execution. Any misuse of these fields can cause AODV to malfunction. Table 2 denotes several vulnerable fields in AODV routing messages and the possible effects when they are tampered.

Table 2: Vulnerable Fields in AODV Packets

Field	Modifications
RREQ ID	Increase to create a new RREQ request.
Hop Count	If sequence number is the same, decrease it to update other nodes' forwarding tables, or increase it to invalidate the update.
IP Headers as well as AODV Source and Destination IP Addresses	Replace it with another or invalid IP address.
Sequence Number of Source and Destination	Increase it to update other nodes' forward route tables, or decrease it to suppress its update.

An attacker could launch a single (packet) attack consisting of several carefully modified fields, or an aggregate attack consisting of multiple attack messages, which cause more damages and last longer than a single attack does. The reader is referred to [1] for a more detailed classification of such attacks (termed atomic and compound attacks) as well as simulations of the impact of such attacks. We will briefly describe some of the attacks below.

3.3 Examples of Single Attacks

3.3.1 Forging Sequence Number

Sequence numbers indicates the freshness of route to the associated node. If an attacker sends out an AODV control packet with a forged large sequence number of the victim node, it will change the route to that victim node. For example, in our example AODV scenario (see Figure 1), if M sends a RREQ, m1, to C with SN.Src equal to 200 (>100), it will take precedence over b1. The route from C to A will go through M instead of going through B. Node M can then control the route between A and D. As another example, if M sends a RREP to B with SN.Dst equal 100 (>61), it will take precedence over c2. B will send data through M to D instead of C; M can then control the route between A and D. This attack can be self-corrected by the protocol when the victim node issues a RREQ or RREP with its sequence number larger than that in the attack packet.

3.3.2 Forging Hop Count

The damage caused by forging of the hop count field will not last as long as the sequence number forging attacks. However, this attack is harder to detect since it is difficult to know the correct hop count to verify the hop count in the attack packet. For example, if M sends a RREQ to C with HC=0(<1) (pretending to be A), it will take precedence over b1 and again, M can control the route. Or, if M sends a RREP to B with HC=0(<1) (pretending to be D) and other values same as c2, it will take precedence over c2 and M can control the route. This attack will be corrected during normal protocol execution when the victim node issues new RREQ or RREP with a higher sequence number. On the other hand, this attack could be very powerful when combined with other attacks to form an aggregate attack as described in the following subsection.

3.4 Examples of Aggregated Attacks

The attacker can combine multiple single attacks to perform a more complicated attack or make the attack last longer. Some interesting attacks are described below.

3.4.1 Man in the Middle Attack

The attacker could issue a fake RREQ and a RREP to poison other node's forwarding table to divert route. The attacker could send a RREQ to C, m1, which is the same as b1 but with higher SN.Src =200 (>100) to take precedence over b1, and send a RREP to B, m2, which is the same as c2 but with SN.Dst=100(>61) in order to take precedence over c2. The next hub of reverse route of C is M instead of B so D and C will go to A through M. The next hub of forward route of B is M instead of C so A and B will go to D through M. Then M could forward the diverted packets from B and C. Therefore, the complete route is ABMCD instead of ABCD

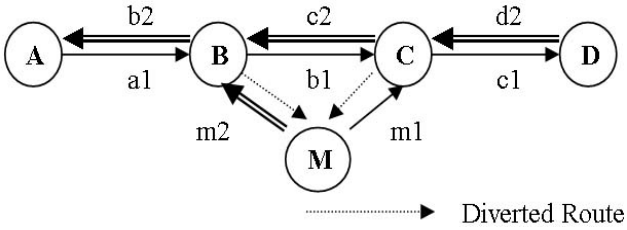


Figure 2: Man in the Middle Attack

3.4.2 Tunneling Attack

Tunneling attack is done by two cooperating malicious nodes that falsely represent the length of available paths by building a tunnel between them. In this way, the malicious nodes can force traffic to route through them.

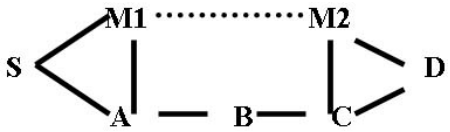


Figure 3: Tunneling Attack

As shown in figure 3, there is no direct link between M1 and M2, but M1 and M2 can pretend to be directly adjacent by tunneling. M1 encapsulates the message and sends it through A, B and C to M2, and falsely claim there is a direct link between M1 and M2. In AODV, when S broadcasts RREQ to A and M1, it will get RREP from A and M1, where their path are {S, A, B, C, D} and {S, M1, M2, D}. S will choose {S, M1, M2, D} but it is actually {S, M1, A, B, C, M2, D}. M1 and M2 successfully prevent S from choosing the really shortest path, {S, A, B, C, D}. Even a cryptography-based solution, such as ARAN [15], cannot prevent this kind of attack.

4. SPECIFICATION-BASED MONITORING OF AODV

Specification-based monitoring compares the behavior of objects with their associated security specifications that capture the correct behavior of the objects. The specifications are usually manually crafted based on the security policy, functionalities of the objects, and expected usage. Specification-based detection does not detect intrusions directly – it detects the effect of the intrusions as run-time violation of the specifications instead. As the specifications are concerned with the correct behavior of objects, specification-based detection does not limit itself to detecting just known attacks. The specification-based detection approach has been successfully applied to monitor security-critical programs [8], applications, and protocols [7]. In particular, specifications for the Address Resolution Protocol (ARP) and the Dynamic Host Configuration Protocol (DHCP) have been used to detect attacks that exploit vulnerabilities in these protocols.

In general, a specification for a network protocol constrains the messages exchanged by the network nodes. The specifications could restrict the way the messages are exchanged (e.g., an ACK followed by a SYN), the contents of the messages. The specifications could also be derived from some desirable global invariants about the protocol.

In applying the specification techniques to monitor AODV, we focus first on the routing messages that are exchanged in the discovery of routes. In particular, we attempt to monitor all the RREQ and RREP messages in a request-reply flow from a source node to a destination node and back to the source. Our specification requires that all nodes send RREQ and RREP messages according to the protocol specifications, and the hop count, RREQ ID, and the sequence numbers are correct.

In the following subsections, we describe how to monitor a request-reply flow using distributed Network Monitors (NM).

4.1 Basic Assumptions

In order to narrow the scope of the problem, we employ the following assumptions. Future investigation of the problem will relax some of the assumptions.

1. The MAC addresses and IP addresses of all mobile nodes are registered in network monitors and remain unchanged.
2. MAC addresses cannot be forged.
3. Every network monitor and its messages are secure and authenticated.
4. Every node must forward or respond to the messages according to the protocol within some finite period of time.
5. Network monitors are well selected to be able to cover all nodes and perform all required functionality.
6. If a node is out of range of a network monitor, it must be in the range of neighboring monitors.
7. If some nodes do not respond to broadcast messages, this will not cause serious problems.

4.2 Run-time Monitoring of Request-Reply Flow

The nature of ad hoc networks prohibits any single IDS node to observe all messages in a request-reply flow. Therefore, tracing of RREQ and RREP messages in a request-reply flow have to be performed by distributed network monitors (NMs).

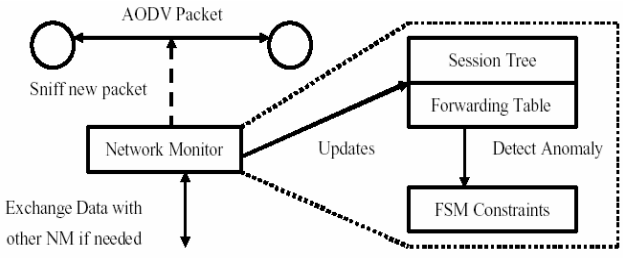


Figure 4: Architecture of Network Monitor

Figure 4 depicts the architecture of a network monitor. Networks monitors passively listening to AODV routing message and detect incorrect RREQ and RREP messages. Messages are grouped based on the request-reply flow to which they belong. A request-reply flow can be uniquely identified by the RREQ ID, the source and destination IP addresses. A RREQ or RREP message can map to a request-reply flow based on these fields as shown below.

RREQ: AODV Source address and RREQ ID

RREP: AODV Source and Destination address

A network monitor keeps track of the RREQ and RREP message last received by each monitored node and maintains the forwarding table of each monitored node. In addition, as each request-reply flow could have several branches – RREQ is a broadcast message and more than one neighbor could continue broadcasting it – NM maintains a session tree to trace the branches. When NM sees an AODV packet as a current packet, NM searches the session tree to find the previous packet of that packet. If NM cannot find the previous packet to match the current packet in the session tree, it will ask its neighboring NMs to find the previous packet. If one of the neighboring NM answers, NM receives the information of the previous packet and the tree it belongs to. Otherwise, NM will treat it as an "Active forge" anomaly. After comparing the current and previous packet, NM inserts the current packet into the session tree for the next current packet. If it is RREP message, NM will mark the new link as red link. Besides, NM will also update its forwarding table. By tracing the session tree, NM can easily match the current and previous packet to detect anomaly, especially in RREQ. Moreover, NM can detect incorrect hop counts and their previous nodes in RREQ. NM can also identify the broken links of corresponding RERR so that it can mark out the broken links and tell its nodes not to use those links in a period of time. Even NM could mark out the node suffering from poor connection and issuing lots of RERR.

Bandwidth overhead is generated by NMs when it needs to ask its neighboring NMs for the information of the nodes which are out of the range of its radio range. This happens when the nodes move out of the range of the NM, or the packet is forwarded to a node that is out of its range.

4.3 Finite-state Machine Constraints

A network monitor employs a finite state machine (FSM) for detecting incorrect RREQ and RREP messages. It maintains a FSM for each branch of a request-reply flow. A request flow starts at the Source state. It transmits to the RREQ Forwarding state when a source node broadcasts the first RREQ message (with a new REQ ID). When a forwarded broadcasting RREQ is detected, it stays in RREQ Forwarding state unless a corresponding RREP is detected. Then if a unicasting RREP is detected, it goes to RREP Forwarding state and stays there until it reaches the source node and the route is set up. If any suspicious fact or anomaly is detected, it goes to the suspicious or alarm states.

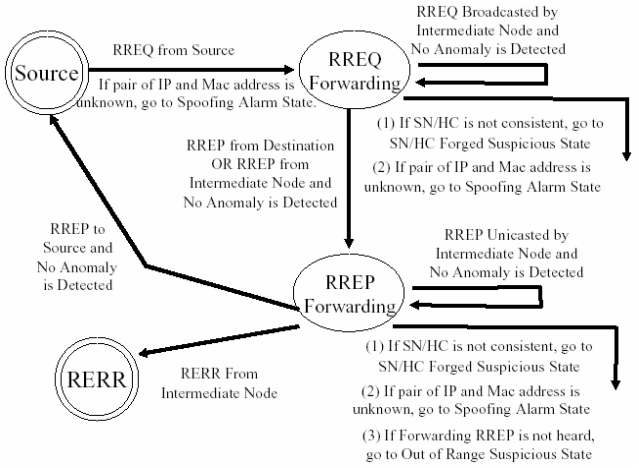


Figure 5: Normal State Diagram

When a NM compares a new packet with the old corresponding packet, the primary goal of the constraints is to make sure that the AODV header of the forwarded control packets is not modified improperly. If an intermediate node responds to the request, the NM will verify this response from its forwarding table as well as with the constraints in order to make sure that the intermediate node does not lie. In addition, the constraints are used to detect packet drop and spoofing.

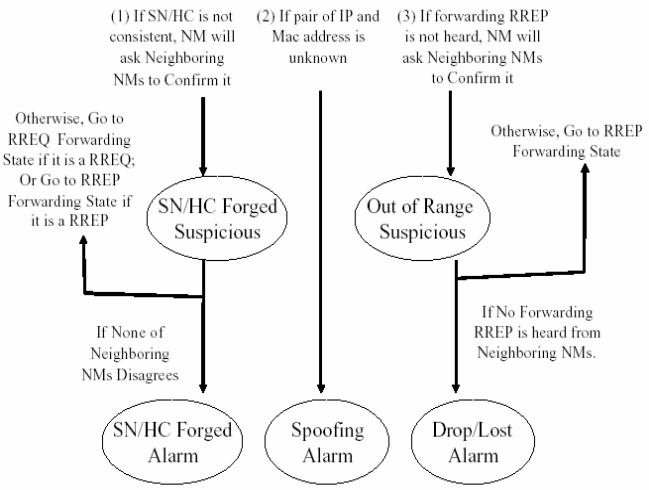


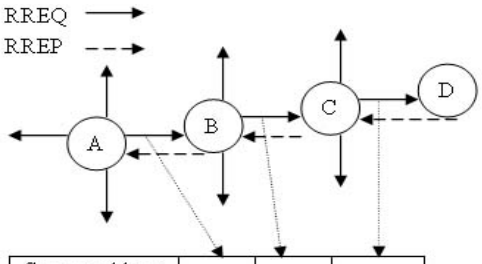
Figure 6: Suspicious and Alarm State Diagram

Figure 6 shows the suspicious and alarm states. If either sequence number (SN) or hop count (HC) is not consistent, it goes to SN/HC Forged Suspicious and NM will ask neighbor NMs to confirm it (Shown as (1)). If none of them disagrees, the request flow goes to SN/HC Forged Alarm. Otherwise, it goes to RREQ Forwarding State if it is RREQ, or it goes to RREP Forwarding State if it is RREP. Out of Range Suspicious state is only applied for RREP and NM will also ask neighbor NMs to confirm it (Shown as (3)). If no disagreement, it goes to Drop/Lost Alarm. Otherwise, it goes to RREP Forwarding state. If the IP and MAC address mapping is unknown, it goes to Spoofing alarm (Shown as (2)). Each branch of a request flow is independent and will be treated separately.

4.4 Matching Current and Previous Messages

To determine the validity of a message (sent by a node, say A), a network monitor needs to identify the corresponding incoming message to A.

For unicast messages, such as RREP, a NM can map current and previous packets easily by looking their source and destination addresses in IP headers. However, in broadcast messages, such as RREQ, the destination address will always be the broadcast address (255.255.255.255). To keep track of the RREQ path, we add one more field to AODV, called previous node (PN). This field indicates the node that previously forwarded the RREQ to the current node.



Source address In IP header	A	B	C
Hop Count	0	1	2
Previous node	--	A	B

Figure 7: Example of Previous Node

For example, in the scenario described in Section 3 (Figure 1), the RREQ message broadcasted by A is forwarded from B to C then to D. Given the previous node field, we can identify the intermediate path AB by the RREQ message sent by B and the path BC by the RREQ message sent by C. The NM knows D responds to this request to C by source and destination address in the IP header of RREP from D. Now, the NM can know that A's request is forwarded by B, C and responded to by D, and therefore have a complete request path from A to D. Also, the NM can know the response path from D to A by the source and destination addresses of the IP header of the unicast RREP messages. Therefore, the NM can trace the complete request flow from A to D and from D back to A.

4.4.1 The Need for Previous Node Field

When NM hears a RREQ with PN, it is able to update the next hop of the reverse route in the forwarding table regarding to PN in RREQ. Otherwise, NM is not able to detect the following two attacks:

(1) A malicious node forwards a RREP to the node that is not the next hop of the reverse route.

(2) If a node, M, forwards RREP to A, but A does not forward it to S, then NM cannot determine if:

a. The destination, A, dropped the packet, or

b. M told the fake smaller hop count in the RREQ it forwarded and M forwards RREP to A via the reverse route it claimed but actually A is always out of M's radio range. In order to achieve this attack, M has to know the network topology near by M and claims a shorter reverse route that is actually invalid.

In (1), with PN, NM can know the next hop of reverse route and therefore can detect a malicious node forwarding packets to the wrong place. In (2), NM could mark out the link between A and M as a bad link. When S rebroadcasts RREQ, D gets a RREQ from M with PN=A, and it will ignore this RREQ. Without PN, D would not know RREQ sent by M was sent from A or some other nodes. Hence D will either ignore all RREQs from M resulting in false negatives, or accept all of them resulting in false positives.

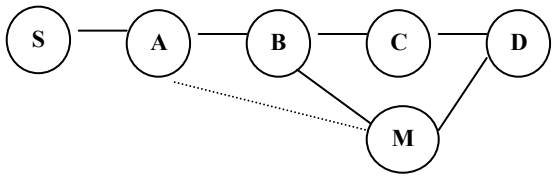


Figure 8: Example Scenario

4.4.2 Functionality of NM

NMs passively listen to wireless media to monitor AODV packets. They exchange information through a secure channel, and only when additional information of nodes is needed, for example, when the session path moves across multiple NM's radio ranges. Moreover, based upon the AODV control messages heard, a NM stores the expected forwarding tables of the nodes within its radio range in order to be able to examine in the future if the nodes are misbehaving. With the low overhead and memory storage, NMs are able to detect system errors and anomalies that could lead to potential (and possibly unknown) attacks in real time with low false positives by employing predefined finite state machine constraints (see below).

4.5 Construction and Processing of Session Trees

Procedure 1 below describes the process at each Network Monitor (NM). Each NM listens to the channel and start processing when it hears a message M being sent within its radio range.

Procedure 1: Network Monitoring Procedure

```
1.  while (true)
2.    wait (until NM hears message M being sent into channel)
3.      if (MacIPUnMatch(M))
4.        DetectSpoofing(M)
5.      else if (M.Type = RREQ)
6.        AddSessionTree (M)
7.      else if (M.Type = RREP)
8.        PeocessSessionTree(M)
9.      else if (M.Type = RERR)
10.       MarkLinkBroken(M)
```

4.5.1 Detect Spoofing

Since each NM has a complete mapping between the Mac address and IP address of every node in the network, a NM can examine M to determine if the Mac-IP address is consistent with the preconfigured data in order to detect the spoofing attack (lines 3, 4).

4.5.2 Monitoring RREQ – Building Session Trees

Procedure 2: AddSessionTree(M)

```
1.  RetrieveTree(M.AODVSrc, M.RRID, SessionTreeList, T)
2.  RetrivePrevMsg(M.PrevNode, PrevM)
3.  CheckConsistency(M, PrevM)
4.  AddTreeNode(M, T)
5.  UpdateForwardingTable(M,F)
```

If M is a RREQ, the NM employs AddSessionTree(M) described in procedure 2. SessionTreeList is the list of trees in which each tree corresponds to each RREQ session. In RetrieveTree procedure (Line 1), AODV source address (AODVSrc) and RREQ ID (RID) in the RREQ are used to identify and retrieve the session tree. If M.IPSrc(Source IP address in IP header of message) is equal to M.AODVSrc(Source IP address in AODV), it indicates that a node has initiated a new RREQ request; so a new session tree will be created. If it cannot retrieve a tree, the NM will request one from its neighboring NMs. If none of them can find a corresponding session tree, an active forged RREQ anomaly is detected.

In RetrivePrevMsg procedure (Line 2), the NM searches the RREQ message (PrevM) that is forwarded right before the current RREQ message (M) according to M's previous node field (M.PrevNode) in the session tree. If the NM and its neighboring NMs fail to find one, it means that the previous node field given in M is incorrect and a fake previous node anomaly is detected. Otherwise, in CheckConsistency procedure (Line 3), the NM verifies values in M such as SN, HC correspond to those in PrevM. Then, the NM trusts the values in M, adds it into the session tree (Line 4) and updates the forwarding table (F) (Line 5) according to the reverse route given in M.

4.5.3 Monitoring RREP

Procedure 3: ProcessSessionTree(M)

1. RetrieveTree(M.AODVSrc,M.AODVDst,SessionTree List,T)
2. **if** (InitRREP(M,T) **and** NotDst(M,T))
3. VerifyRREP(M,F)
4. **else if** (ForwardedRREP(M,T))
5. RetrivePrevMsg(M.IPSrc, PrevM)
6. CheckConsistency(M, PrevM)
7. AddRREPPath(M, T)
8. UpdateForwardingTable(M,F)

If M is an RREP, the NM processes M in ProcessSessionTree(M) (shown in Procedure 3). In RetrieveTree procedure (Line 1), the AODV source address (AODVSrc) and AODV destination address (AODVDst) in RREP are used to identify and retrieve the session tree. If the NM and its neighboring NMs fail to get one, an active forged RREP is detected.

InitRREP (Line 2) is true if a node (M.IPSrc) that is not in the tree replies a RREP to one of the node (M.IPDst) in the tree. NotDst is true if the sender (M.IPSrc) is not the destination of the request (M.AODVDst). The NM will only verify a new RREP generated by an intermediate node according to its forwarding table since NMs trust new RREP issued by the destination of AODV request. ForwardedRREP is true if the sender of the RREP is the tail of RREP path and the destination of the RREP is not in the RREP path but in the session tree. Then the NM retrieves the previous message (PrevM) which is the tail of RREP path and check consistency according to PrevM. Now NM trusts this new RREP, adds it into the RREP path of the tree, and updates the forwarding table (F) according to the forwarding route given in M. In addition, if all RREP paths go back to the source of the request (M.AODVSrc) and no more RREPs are detected, then the whole tree can be discarded. Also, before a complete RREP path to source is established, if no RREP is added in a period of time, the NM will report a drop/loss anomaly.

4.5.4 Monitoring RERR

Finally, if M is an RERR, then the NM updates the forwarding table according to which node is unreachable by which node. To prevent an attacker from repeatedly using RERR to perform an attack, a broken link is forced to remain in that state for a finite period of time.

5. EXAMPLES

In order to show how the IDS detects attacks, we first describe how the network monitors trace AODV packets based on the AODV scenario in Section 3.1. Then we show how we detect the single attacks in Section 3.3 and aggregated attacks in Section 3.4.

5.1 Tracing AODV Packets

In Figure 9, two network monitors, N1 and N2, work cooperatively and trace the request flow shown in section 3.1. Table 3 shows the AODV packets that N1 and N2 see in each time slot. Table 4 shows how N1 and N2 build up their session trees step by step according to the AODV packets shown in table 3. At time slot 2, N2 sees b1 but did not see the original packet sent from A, so N2 asks its neighboring monitor, N1, to confirm this. Similarly, at time slot 5, N1 sees c2 and asks N2 to retrieve the complete session tree. Tables 5 and 6 show the forwarding tables of N1 and N2 according to AODV packets they see in each time slot.

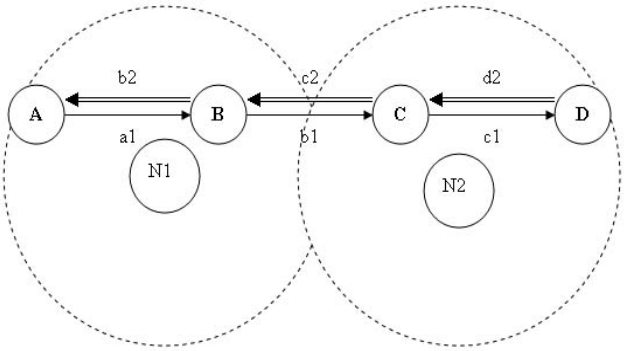


Figure 9: Example AODV Scenario with Network Monitors

Table 3: Packets Seen by NM in Each Time Slot

Time	1	2	3	4	5	6
N1	a1	b1			c2	b2
N2		b1	c1	d2	c2	

Table 4: Session Tree built by NM in Each Time Slot

(— : RREQ only; = : RREQ and RREP)

Time	1	2	3	4	5	6
N1	A	A-B	A-B	A-B	A-B, Ask N2	A=B=C=D
N2		A-B, Ask N1	A-B-C	A-B-C=D	A-B=C=D	A=B=C=D

Table 5: Entries in Forwarding Table

	Dst	Next	SN	HC
a1	A	A	100	1
b1	A	B	100	2
c1	A	C	100	3
b2	D	B	61	3
c2	D	C	61	2
d2	D	D	61	1

Table 6: Forwarding Table in N1 and N2

(Time Slot in Parentheses)

N1: (Time slot)		N2: (Time slot)	
A	B	C	D
	a1(2)	b1(3)	c1(4)
b2(6)	c2(5)	d2(4)	

5.2 Detecting Simple Attacks

5.2.1 Detect Attacks by Forged Sequence number

According to the forwarding table in N1 and N2— SN.Src = 100 and SN.Dst=61. If N1 or N2 detect any packet having SN that is larger than it should be and that packet is not sent by the owner of SN (IP.Src not equal to source or destination Node (depending on message being RREQ or RREP)), it will treat it as an attack. Therefore, the attacks shown in section 2.4 will be detected.

5.2.2 Detect Attacks by Forged Hop count

According to the forwarding table and session tree, if the hop count does not increase by 1 following the session tree, NM will treat it as an attack. Therefore, the attacks shown in section 2.4 will be detected.

5.3 Detecting Aggregated Attacks

5.3.1 Man in the middle attack

Since SN of the packets sent by M is larger than that NMs have and the packets were not sent by the owner of SN, (IP.src not equal to source or destination Node (depending on message being RREQ or RREP)) the NM will detect the attack.

5.3.2 Tunneling attack

In this attack, the attack claims that the route is {S, M1, M2, D} although the real route is {S, M1, A, B, C, M2, D}. When M2 gets the unicasting RREP which is actually forwarded to C, our IDS would know it by checking its IP header and notice that it is not forwarded by M1 according to the route given by the AODV packets sent by M1 and M2. Therefore, our IDS detects that the link between M1 and M2 is actually fake.

6. FUTURE WORK

The focus of our future work is threefold. We briefly describe these below.

6.1 Profiling Normal Network QoS

When the IDS detects that a unicast packet is lost, it cannot distinguish between lost or dropped packets because the probability of packet loss is much higher in wireless networks. Besides, in ad-hoc networks, since routing errors occur frequently, it is difficult to judge if an error message is real or fake. So we need to profile packet loss, packet error and packet generation in order to define a reasonable error ratio and reduce false positives for the IDS. Therefore, if the IDS detects that the number of error messages, packet drops, and packet generation for a particular node in a certain period of time is higher than the threshold according to the current profile, there is higher confidence in labeling the anomaly as an attack. In other words, profiles that indicate the current QoS of the network will assist in refining the operation of the IDS.

6.2 Using P2P to refine the NM architecture

P2P networks serve as an attractive model for the distribution of cyber-security information. Commonly used P2P networks, however, have serious drawbacks however. These simple P2P protocols are not naturally scaleable; large numbers of common queries lead to redundant message floods to all users. Caching and aggregate request forwarding can help to alleviate this problem. We envision a P2P based network for the Network Monitors presented in this paper. Each member of a very large cooperative network can potentially access information they deem relevant from all other nodes. Individual NMs simultaneously monitor their local environment for intrusions and query other cooperating members for occurrences of events in which they are interested. In this manner, local behavior can be correlated with activity witnessed by remote devices, allowing for the recognition of widespread attack behavior or sophisticated distributed coordinated attack with no central correlation nodes.

6.3 Experimentation and Verification

We will simulate aspects of intrusive behavior of malicious hosts employing the *ns-2* network simulator. We will be considering simulation environments that involve dense, complex networks with high node mobility and substantially dynamic topologies. We are interested in a performance analysis of this approach in the presence of both node mobility and dynamic topologies as well as under specific node failure/link failure scenarios.

7. CONCLUSION

We propose a specification-based intrusion detection system that can detect attacks on the AODV routing protocol. In a specification-based intrusion detection approach, the correct behaviors of critical objects are manually abstracted and crafted as security specifications, and this is compared with the actual behavior of the objects. Intrusions, which usually cause object to behavior in an incorrect manner, can be detected without exact knowledge about them. This approach can, thus, address unknown attacks as well. The IDS presented in this paper is built on a distributed network monitor architecture that traces AODV request-reply flows. Network monitors audit every RREQ, RREP and RERR in order to build and update complete request-reply session trees and corresponding forwarding tables. Constraints on the request-reply flow are specified using finite state machines. We describe procedures for constructing and processing the session trees, and present examples of detecting attacks successfully. This research is the first effort to apply specification-based detection techniques to detect attacks in the routing within ad hoc networks. We illustrate that our algorithm can effectively detect most of the serious AODV routing attacks effectively, and with low overhead.

8. ACKNOWLEDGMENTS

This research is supported in part by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program.

9. REFERENCES

[1] Peng Ning, Kun Sun, "How to Misuse AODV: A Case Study of Insider Attacks against Mobile Adhoc Routing Protocols," in Proceedings of the 4th Annual *IEEE Information Assurance Workshop*, pages 60-67, West Point, June 2003.

[2] S. Bouchegger and J. –Y. L. Boudec. Performance Analysis of the Confidant Protocol. In Proceedings of the 3rd ACM Symposium on Mobile Ad Hoc Networking and Computing. Pp 226-236, 2002.

[3] L. Buttyán and J.-P. Hubaux, “Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks Technical Report No. DSC/2001/046, Swiss Federal Institute of Technology, Lausanne, August 2001

[4] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. “Ariadne: A Secure On-Demand Routing Protocol for AdHoc Networks. In Proceedings of the Eighth ACM Intl. Conf. on Mobile Computing and Networking (MobiCom ’02), ACM, Atlanta, Sept. 2002, pp 12-23.

[5] K. Ilgun, R. Kemmerer, and P. Porras , “State Transition Analysis: A Rule-based Intrusion Detection Approach”, IEEE Transactions of Software Engineering, 2(13):181-199, March 1995.

[6] H. Javitz and A. Valdes, “The NIDES Statistical Component Description and Justification,” Technical Report, Computer Science Laboratory, SRI International, Menlo Park, CA , Mar 1994.

[7] C. Ko, P. Brutch, J. Rowe, J., et al. 2001. System Health and Intrusion Monitoring Using a Hierarchy of Constraints. In Proceedings of the 4th Symposium on Recent Advances in Intrusion Detection. Davis, CA.

[8] C. Ko, M. Ruschitzka and K. Levitt, “Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specification-based Approach,” In Proceedings of the 1997 IEEE Symposium on Security and Privacy, May 1997.

[9] O. Kachirski, R Guha, “Effective Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks”, 36th Annual Hawaii International Conference on System Sciences (HICSS'03), January 06 - 09, 2003

[10] U. Lindqvist and P. Porras, “Detecting Computer and Network Misuse Through the Production-Based Expert System Toolset (P-BEST)”, In Proceedings of the 1999 Symposium on Security and Privacy, May 1999.

[11] S. Marti, T. Giuli, K. Lai and M. Baker. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000), August, 2000.

[12] P. Michiardi, R. Molva, “Core: A Collaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks”, Communication and Multimedia Security 2002 Conference.

[13] R. Rao and G. Kesidis, “Detection of malicious packet dropping using statistically regular traffic patterns in multihop wireless networks that are not bandwidth limited”, Brazilian Journal of Telecom-munications,2003

[14] R. Ramanujan, S. Kudige, T. Nguyen, S. Takkella, and F. Adelstein, “Intrusion-Resistant Ad Hoc Wireless Networks”, Proceedings of MILCOM 2002, Oct. 2002.

[15] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields and E. Belding-Royer. A Secure Routing Protocol for Ad Hoc Networks. In Proceedings of IEEE ICNP, 2002.

[16] S. Marti, T. Giuli, K. Lai and M. Baker. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000), August, 2000.

[17] L. Zhou and Z. J. Haas. Securing ad hoc networks. In IEEE Networks, 13(6):24-30, 1999.

[18] M. Zapata and N. Asokan, Securing Ad hoc Routing Protocols, 2002.

[19] Y. Zhang and W. Lee. Intrusion Detection in Wireless Ad Hoc Networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000), August, 2000.

[20] E. Belding-Royer, Private Communication, 2003.