Secure Routing and Intrusion Detection in Ad Hoc Networks *

Anand Patwardhan, Jim Parker and Anupam Joshi
Computer Science and Electrical Engineering Department
University of Maryland at Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250
{anand2, jparke2, joshi}@cs.umbc.edu

Michaela Iorga and Tom Karygiannis
National Institute of Standards and Technology
100 Bureau Drive, Gaithersburg, MD 20899
{miorga, karygiannis}@nist.gov

Abstract

Numerous schemes have been proposed for secure routing protocols, and Intrusion Detection and Response Systems, for ad hoc networks. In this paper, we present a proof-of-concept implementation of a secure routing protocol based on AODV over IPv6, further reinforced by a routing protocol-independent Intrusion Detection and Response system for ad-hoc networks. Security features in the routing protocol include mechanisms for non-repudiation, authentication using Statistically Unique and Cryptographically Verifiable (SUCV) identifiers, without relying on the availability of a Certificate Authority (CA), or a Key Distribution Center (KDC). We present the design and implementation details of our system, the practical considerations involved, and how these mechanisms can be used to detect and thwart malicious attacks. We discuss several scenarios where the secure routing and intrusion detection mechanisms isolate and deny network resources to nodes deemed malicious. We also discuss shortcomings in our approach and conclude with lessons learned, and ideas for future work.

1. Introduction

Recent years have witnessed a proliferation of mobile devices. Corporations and government agencies alike are increasingly using embedded and wireless technologies, and working towards mobilizing their workforce. Mobile devices typically support several forms of wireless connectivity like 802.11, IrDA, Bluetooth, etc. Among them, "Converged Mobile devices" – devices with integrated functionality of cell-phones and PDAs, make use of services like GSM and GPRS, for access to the Internet. Due to technology limitations, however, wireless access to the service providing infrastructure is limited to particular areas. Moreover, buildings and other physical obstructions further restrict availability.

Consequently, the productivity of a mobile workforce relying solely on infrastructure-based network services is restrictive and unsatisfactory. Reliable communication is a necessity for nodes in a dense network of independent mobile devices such as, participants in a meeting. Several co-operative mechanisms exist which enable such devices to interact through peer relationships, even in the absence of infrastructure support. Other factors of cost, response time, and efficiency strongly motivate the use of ad hoc networks.

Ad hoc networks, as the name suggests, have no supporting infrastructure. Ad hoc networks are comprised of a dynamic set of cooperating peers, which share their wireless capabilities with other similar devices to enable communication with devices not in direct radio-range of each other, effectively relaying messages on behalf of others. Conventional methods of identification and authentication are not available, since the availability of a Certificate Authority or a Key Distribution Center cannot be assumed. Consequently, mobile device identities or their intentions cannot be predetermined or verified.

Several routing protocols for ad-hoc networks have been proposed like DSDV [15], DSR [7], AODV [2], TORA [13] etc. A majority of these protocols assume

* This research was supported by NSF award 9875433, and a grant from Booz Allen Hamilton

a trustworthy collaboration among participating devices that are expected to abide by a "code-of-conduct". Herein lie several security threats, some arising from shortcomings in the protocols, and others from the lack of conventional identification and authentication mechanisms. These inherent properties of ad hoc networks make them vulnerable, and malicious nodes can exploit these vulnerabilities to launch various kinds of attacks. To protect the individual nodes and defend the Mobile Ad Hoc Network (MANET) from malicious attacks, intrusion detection and response mechanisms are needed.

Conventional Intrusion Detection Systems(IDS) have relied on monitoring real-time traffic at switches, gateways, and routers. Vulnerabilities in Medium Access Control(MAC) for wired networks have been protected by physical partitioning and restricted connectivity amongst networks. The wireless connectivity of mobile nodes shares a common medium but cannot be partitioned, nor can the mobility of the nodes be restricted. Mobility introduces additional difficulty in setting up a system of nodes cooperating in an IDS. A node's movements cannot be restricted in order to let the IDS cooperate or collect data and a node cannot be expected to monitor the same physical area for an extended period of time. A single node may be unable to obtain a large enough sample size of data to accurately diagnose other nodes.

Several architectures and detection mechanisms for IDS for MANETs have been proposed so far. Simulations and illustrations have been used to validate the feasibility of proposed schemes for secure routing and intrusion detection. However, to the best of our knowledge, our combination of a secure routing protocol and IDS is the first actual implementation. We present a detailed analysis of issues involved in the implementation and deployment of a secure routing protocol and an IDS in our testbed. We present interesting results that provide insights into practical considerations in such a deployment that have not been addressed thus far.

In this paper we describe our implementation of a Secure routing protocol, SecAODV. We also provide a description of the IDS module. Furthermore, we discuss several other routing protocols proposed in the literature, in the related work section. For the Secure AODV (henceforth referred to as SecAODV) we have adapted the AODV implementation by Tuominen [21], and added security features to it, which have been previously proposed in [1, 20] . We further enhanced the security of our MANET testbed by deploying a stateful packet snooping Intrusion Detection System(IDS) based on an algorithm proposed in our previous work [14]. SecAODV and the Snooping IDS complement each other in being able to detect most of the prevalent attacks. Our goal is to detect malicious or chronically faulty nodes and deny them network resources. We describe different kinds of security threats in pervasive environments. We then describe the design and implementation of SecAODV and IDS, and discuss how this combination protects benign nodes in the MANET. We conclude with a discussion on lessons learned in our implementation, feasibility of proposed methods, and ideas for future research.

2. Background and Related Work

2.1. Secure Routing Protocols

As previously mentioned, a majority of the routing protocols proposed in the literature assume non-hostile environments. MANETs are extremely vulnerable to attacks due to their dynamically changing topology, absence of conventional security infrastructures and open medium of communication, which, unlike their wired counterparts, cannot be secured. To address these concerns, several secure routing protocols have been proposed: SAODV [23], Ariadne [5], SEAD [4], CSER [8], SRP [12], SAAR [22], BSAR [1], and SBRP [20].

Our implementation of the SecAODV is based upon the protocol proposed in BSAR [1] and SBRP [20] for DSR. This solution is a highly adaptive distributed algorithm designed for IPv6-based MANETs that do not require:

- prior trust relations between pairs of nodes (e.g. a trusted third party or a distributed trust establishment)
- time synchronization between nodes, or
- prior shared keys or any other form of secure association

The protocol provides on-demand trust establishment among the nodes collaborating to detect malicious activities. A trust relationship is established based on a dynamic evaluation of the sender's "*secure IP*" and signed evidence, contained in the SecAODV header. This routing protocol enables the source and destination nodes to establish a secure communication channel based on the concept of "*Statistically Unique and Cryptographically Verifiable*" (SUCV) identifiers [1, 10] which ensure a secure binding between IP addresses and keys, without requiring any trusted CA or KDC. The concept of SUCV is similar to that of Cryptographically Generated Address (CGAs) [17]. SUCVs associate a host's IPv6 address with its public key that provides verifiable proof of ownership of that IPv6 address to other nodes.

2.2. Intrusion Detection Schemes

MANETs present a number of unique problems for Intrusion Detection Systems (IDS). Differentiating between malicious network activity and spurious, but typical, problems associated with an ad hoc networking environment is a challenging task. In an ad hoc network, malicious nodes may enter and leave the immediate radio transmission range at random intervals or may collude with other malicious nodes to disrupt network activity and avoid detection. Malicious nodes may behave maliciously only intermittently, further complicating their detection. The loss or capture of unattended sensors and personal computing devices may allow for a malicious node to obtain legitimate credentials and launch more serious attacks. A node that sends out false routing information could be a compromised node, or merely a node that has a temporarily stale routing table due to volatile physical conditions. Dynamic topologies make it difficult to obtain a global view of the network and any approximation can become quickly outdated.

Traffic monitoring in wired networks is usually performed at switches, routers and gateways, but an ad hoc network does not have these types of network elements where the IDS can collect audit data for the entire network. A wired network under a single administrative domain allows for discovery, repair, response, and forensics of suspicious nodes. A MANET is most likely not under a single administrative domain, making it difficult to perform any kind of centralized management or control. Network traffic can be monitored on a wired network segment, but ad hoc nodes or sensors can only monitor network traffic within its observable radio transmission range.

Zhang and Lee [24] categorize host-based IDSs based on anomaly detection and misuse detection. Anomaly detection-based systems detect intrusions based on an established baseline of normal behavior. Misuse detection involves identifying attack signatures and usage patterns associated with known attacks. They point out that unlike wired networks there are no fixed "concentration points" where real-time traffic monitoring can be done; audit collection is limited by radio-range of the devices. Also, communication patterns are different from wireline devices and mobile devices are often expected to operate in disconnected mode. Anomalies are not easily distinguishable from localized, incomplete, and possibly outdated information. So, anomaly detection schemes are not directly applicable in wireless ad hoc networks. Hence, they propose a new architecture for an IDS, based on IDS agents.

Other proposals include use of mobile agents trained to detect intrusions [] and specification based algorithms [19]. The performance costs and security risks associated with these approaches, however, limit their practical uses.

Cheng et al. [19] describe several attacks possible in the base AODV protocol. They illustrate the use of a finite state machine to detect anomalous behavior in order to determine attacks. They also suggest the use of an additional previous hop field to ascertain the source/path of AODV control messages.

Our approach to intrusion detection is similar to that proposed by Zhang and Lee [24]. We deploy IDS monitors on individual nodes for detecting intrusions within radio range. We consider these local monitors as building blocks for further work on collaborative IDS schemes for MANETs.

3. Security Threats

Attacks can be targeted at the routing protocol in which the malicious node actively disrupts the functioning of the cooperative routing mechanisms. A secure routing protocol is intended to minimize or prevent the impact of possible attacks against nodes in a MANET.

In general, the attacks can be classified as:

I. Routing disruption attacks

II. Resource consumption attacks

III. Attacks on data traffic

3.1. Routing Disruption attacks

In a "routing disruption attack," a malicious node intentionally drops control packets, misroutes data, or disseminates incorrect information about its neighbors and/or its pre-discovered routing capabilities to particular destinations. An attacker might try to:

(i) forge messages by spoofing originator or destination addresses,

(ii) signal false route errors or modify route error messages,

(iii) alter or replace originator, destination or sender addresses in routed messages.

3.2. Resource Consumption attacks

In a "*resource consumption attack*" also known as "*resource exhaustion attacks*," an attacker might try to consume network resources by:

(i) initiating large number of route requests to bogus destinations in order to exhaust the resources of the network, or

(ii) playing the "*gray hole attack*" or "*selective dropping*" of packets, resulting in increased number of route requests from neighbor nodes that have limited routing capabilities, exhausting neighbors' resources.

3.3. Colluding adversaries

A group of malicious nodes can collude in attacking the network causing far more damage than a single node. In general, if keying material is compromised or a malcicious node colludes with others to intentionally disrupt communications, the extent of damage increases with the number of colluding adversaries and the availability of keying material.

Several typical attacks against MANETs have been identified in the literature as follows:

(i) The "*Wormhole attack*." An adversary listens to a message in one part of the network, and replays it in another part of the network with the help of another colluding, malicious node. Wormhole attacks can be classified under colluding adversaries that have cryptographic key material.

(ii) The "*Invisible-node attack*." This attack can be launched by any node in the routing path. It can be considered as a *man-in-the-middle* attack. The damage caused by this attack is limited to the path on which the node is present and it can be classified under non-colluding adversaries attack.

(iii) The "*Rushing attack*." This attack can be launched against any protocol that implements a suppression function for duplicate packets (i.e., duplicate packet detection and suppression) or some kind of waiting time. The damage caused by this attack depends on the protocol under question. In this attack, an adversary rushes a spurious packet to a destination (possibly to an intermediate node on the path or to a destination) making the legitimate packet look like a duplicate. Thus, the legitimate packet is discarded. The technique of duplicate suppression is usually used to make routing based on network flooding efficient. More efficient, non-flooding methods will render this attack harmless.

4. Prototype implementation details

4.1. Hardware used and testbed description

4.2. Assumptions and observations

- Interfaces have a promiscuous mode to monitor traffic

Handheld Device	iPAQ 3800 Series
Processor	206 MHz Intel StrongARM SA-1110 32-bit RISC Processor
Memory	64 MB SDRAM, 32 MB flash ROM Memory
Wireless access	Orinoco and Cisco Aironet 802.11b cards with wireless sleeves

Table 1. iPAQ 3800 Series Specifications

- Key lengths are sufficiently long, making it infeasible to compute or guess a private key knowing only the public key, but on the other hand do not make signature computation and verification computationally expensive for the mobile device
- Normal packet drop rates can be dynamically determined and thresholds established to distinguish malicious behavior from trustworthy conduct.

We do not, however, require MAC addresses to be unforgeable, since the SUCV identifiers provide secure bindings between IPv6 addresses and public keys. Identity is not determined by MAC addresses alone. Spoofing of IPv6 addresses and MAC addresses can be detected, since signature verification will fail unless private keys have been compromised. A malicious node may change its own MAC address and IPv6 address periodically to evade detection. Thus, to go undetected, the attacker will need to change their IPv6 address very often, and incur the additional expense of computing a SUCV identifier every time. Consequently such an attack is largely ineffective, and quite expensive for the attacker.

4.3. SecAODV

4.3.1. Overview The SecAODV implements two concepts which are common features in both BSAR [1] and SBRP [20]:

- Secure binding between IP version 6 (IPv6) addresses and the RSA key generated by the nodes themselves, and independent of any trusted security service, and
- Signed evidence produced by the originator of the message and signature verification by the destination, without any form of delegation of trust

IPv6 was adopted for its large address space, portability and suitability in generating SUCVs. Of special importance is the address auto-configuration feature available in IPv6 that allows IP auto-configuration for the nodes on a need basis.

The implementation follows Tuominen's design [21]. It uses two kernel modules: `ip6_queue` and `ip6_nf_aodv`, and a user space daemon `aodvd`. The `ip6_queue` module is the queuing packet handler. The `ip6_nf_aodv` module decides whether a packet is queued or not. It also manipulates the route lifetime. If the queue handler module is not registered, the packets are dropped. The `aodvd` daemon allows for specific settings from debugging information to configuration file, logging information, etc. For more information and detailed description of the functions can be found in [21].

4.3.2. Secure Address Auto-Configuration and Verification To join a MANET, a node executes a script that sets its Service Set Identifier (SSID) using the `iwconfig` utility. The script then proceeds to install and configure all IPv6 and SecAODV related kernel modules, and finally starts the `aodvd` daemon. The daemon obtains its site and global subnet identifiers, and runtime parameters from a configuration file and/or from the command line. The `aodvd` daemon then generates a 1024-bit RSA key pair. Using the public key of this pair, the securely bound global and site-local IPv6 addresses are generated. To derive the addresses, a node generates a 64-bit pseudo-random value by applying a one-way, collision-resistant hash function to the newly generate, uncertified, RSA public key. However, only 62 bits out of the generated 64 bits will then be used for the IPv6 address because 2 bits of the address space are reserved. The final IP is generated by concatenating the subnet identifier with the pseudo-random value derived from the public key and by setting the 2 reserved bits, according to RFC 3513 (2373) [16]. A source node uses the secure binding to authenticate its IP address to an arbitrary destination. Upon completion of the RSA keys generation and IP address configuration, SecAODV can optionally broadcast "Hello"-type, signed messages to its neighbors (using the multicast address `ff02::1`) to make its presence known. Upon IPv6 address and signature verification, the neighbors update their routing tables with the new information.

4.4. Overview of working of SecAODV over IPv6

The AODV protocol, as proposed by RFC 3561 [2], is comprised of two basic mechanisms, viz. route discovery and maintenance of local connectivity mechanisms. The Route Discovery mechanism is employed in an "Ad Hoc, On Demand" fashion. The source node S - the device that requests communication with another member of the MANET referred to as destination D - initiates the process by constructing and broadcasting a signed route request message `RREQ`. The format of the `RREQ` message differs from the one proposed in [2]. An AODV message contains the RSA public key of the source node S and that it is digitally signed to ensure the node's authentication and message integrity (refer to fig. 1). Upon receiving a `RREQ` message, each node member of the MANET authenticates the source node S and verifies message integrity by checking the IP address using the same secure bootstrapping algorithm described in section 4.3.2, and by verifying the signature against the provided public key. Upon successful completion of the verification process, the node updated its routing table with the source and router IP addresses, if any, and then checks the destination IP address. If the message is not addressed to it, it rebroadcasts the `RREQ`. If the current node is the destination, it constructs a route reply message (`RREP`) addressed to the source node S. The message is signed and it includes the destination's public key as shown in Fig. 1. The destination node D unicasts the `RREP` back to the neighboring node that initially forwarded the `RREQ`. The neighbor address is retrieved from its own routing table, under source address. Upon receiving a `RREP`, any routing node verifies the destination D's IP address and signature against the included public key, updates its own routing table with the destination D and router addresses, if any, and unicasts the message to the router listed in its routing table under the source S address entry. If the route entry to S does not exist or has expired, the message is dropped and an error message is sent to all affected neighbors. If the source node does not receive any reply in a predetermined amount of time, it rebroadcasts new route requests. A detailed explanation of the process can be found in [2]. The *Maintenance of Local Connectivity mechanism* is optionally achieved by periodically broadcasting Hello-type messages. In our implementation these messages are signed and contain the sender's public key for authentication and message integrity verification. Additional information on local connectivity maintenance can be found in [2]. During our implementation and testing of AODV and SecAODV, we observed that the protocol's performance is very sensitive especially to the `HELLO_INTERVAL` and all parameters related to it: `ACTIVE_ROUTE_TIMEOUT`, `DELETE_PERIOD`, `MY_ROUTE_TIMEOUT`, described in [2]. From our experience we learned that the best practice for optimal performance is to set the lifetime of the route entry for the intermediated nodes to the `NET_TRAVERSAL_TIME` plus the local message verification time. In this way, for a well-configured network, operating in an ideal, noise-free environment, the communication between two non-

neighboring nodes can be achieved once and maintained via message exchanges without exhausting Route Discovery requests.

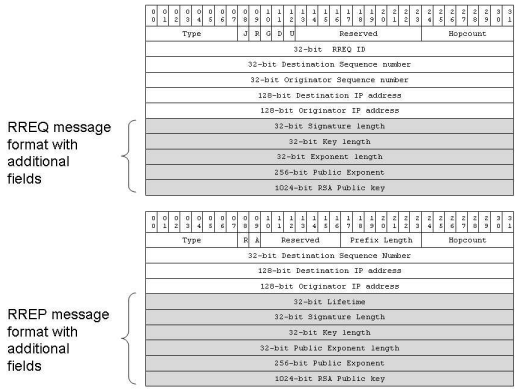

Figure 1. SecAODV message formats

5. Design of IDS

Although encryption and signed headers are intrusion prevention measures, security holes remain nonetheless. An IDS further strengthens the defense of a MANET. A reliable IDS, operating within a MANET, requires that trust be established amongst collaborating nodes in the absence of any pre-existing trust associations, or the availability of an online service to establish such associations. The use of SUCVs is thus well-suited for such situations.

5.1. Design Considerations

Collaborative IDSs will perform best in a densely populated MANET with limited mobility, and will perform worse in a sparsely populated MANET with significant mobility. The effectiveness of a collaborative IDS also depends on the amount of data that can be collected by each node. The longer the nodes are members of the MANET, the greater the availability of meaningful data for further analysis. The degree of mobility of each node in the network will also have a significant impact on its effectiveness. In a MANET with a high degree of mobility, if the number of routing error messages causes by legitimate reasons far exceeds the number of routing error messages caused due to the presence of malicious nodes, the effectiveness or benefit of such an IDS may be minimal. The damage that could be caused by a malicious node in highly mobile environment would, however, also be minimal since malicious routing messages would likely make up a small percentage of routing error messages.

Sensor networks may be less ephemeral and less mobile, while other networks may be characterized by sporadic participation of individual members. MANETs with loose or no prior associations would be more difficult to diagnose than a MANETs comprised of nodes from the same organization with strong associations. Clearly, the latter case would present a more challenging problem. In a network in which nodes have sporadic participation, the damage malicious nodes are likely to cause would also be less serious and more of a nuisance than a serious performance threat. The IDS would perform differently in an open MANET, one in which participation is not restricted, versus a closed MANET, one in which participation is restricted in both number and by the possession of certain credentials.

5.2. Design goals

5.2.1. Scalability Snooping on all packet traffic is prohibitively expensive for most resource-constrained mobile devices, especially when traffic increases as the number of nodes within radio-range increase. In dense networks, there will be a large number of neighbor nodes. Also, as newer wireless standards increase the radio-range of wireless interfaces, resulting larger ranges will have the same effect. The IDS should allow selective processing of packets and ignore the rest. The effectiveness of the IDS will depend on its scalability.

5.2.2. Platform for a collaborative IDS In order to implement a truly robust IDS there will be a need to aggregate data from multiple architectural layers. Alarms and thresholds placed at the network layer can report on the detection of routing misbehaviors such as observed incorrect packet forwarding. The MAC layer may alarm on nodes that send malicious CTS messages designed to deny other nodes network access. The Transport layer may contain signatures for known attacks such as the SYN flood.

Delegating collaboration and Trust issues to the application level, the IDS agent should enable collection of local audit data. The notion of Trust is determined through an aggregation of information collected from multiple observing layers providing input for evaluation algorithms at the Application layer. Collaboration not only comes from within the node, but can be shared between nodes as Trust and reputation values are passed from throughout the network.

5.2.3. Enable protocol specific IDS The IDS should allow monitoring of packet traffic for specific protocols.

Specific protocols behave in a predictable pattern. Intrusion detection makes use of these patterns to spot abnormal behavior and in some instances, specific signatures indicating malicious activity. Some protocols are more likely than others to be used with malicious intent. For example in TCP a SYN flood can use up available ports on the target machine effectively denying service.

5.3. Scope of IDS

In our implementation approach we focus on detecting intrusions based on anomalous behavior of neighboring nodes. Each node monitors particular traffic activity within its radio-range. An audit log of all locally detected intrusions is maintained as evidence of misbehavior. Intrusions are associated with pairs of IPv6 and corresponding MAC addresses. Once local audit data is collected, it can be processed using some centralized/distributed algorithm, to detect ongoing attacks from the aggregated data. Such collective analysis is however subject to trust issues, since the problem of Identification and Authentication remains. Rather in our current implementation, we focus only on the local detection and response part, to provide a foundation for such a collaborative IDS. By virtue of the SUCV identifiers, we can confidently identify the misbehaving nodes and associate intrusions with them. Each node listens to all its neighbor's activities.

5.3.1. Intrusion Detection We detect intrusions by neighboring nodes by their deviation from known or expected behavior. When nodes act as forwarding nodes, offering routes to other destinations, it is expected that those node actually forward data packets, once a route through them is actually setup. Nodes are expected to retransmit the message without modifying the payload towards the intended recipient. We can categorize packet traffic into control packets that exchange routing information, and data packets. Depending on what routing protocol is being used, routing information may or may not be contained in the control packets, e.g. in DSR the routing information is present in the control message itself; AODV on the other hand, does not have such information. Regardless of how routes are actually setup, data packets should not be modified, with the exception of some fields like hopcount in the IPv6 header. A node can thus monitor most of the packet traffic of its neighbors in promiscuous mode, while they are in radio-range. A node receiving packets but not forwarding them can be detected.

We monitor AODV control messages and data stream packets only. We do not monitor control messages for faithful retransmissions. Since control messages are signed by the senders, modifications will be caught in the signature verification at the receiver.

5.3.2. Intrusion Response The purpose of intrusion detection is to isolate misbehaving nodes and deny them network resources. Nodes may be maliciously dropping packets or may have a genuine problem that prevents them from forwarding packets. Chronically faulty or malicious behavior, however, can be distinguished from transient failures by monitoring their activity over a period of time and setting thresholds. Such nodes are then deemed malicious and denied network resources. This can be done in two ways viz. unilaterally ignoring all traffic to or from a malicious node, and calling a vote on other members in the MANET to decide upon the eviction of a suspected node from the MANET [14]. Though this is a design goal, the collective response part has not yet been implemented.

5.4. Stateful packet monitoring

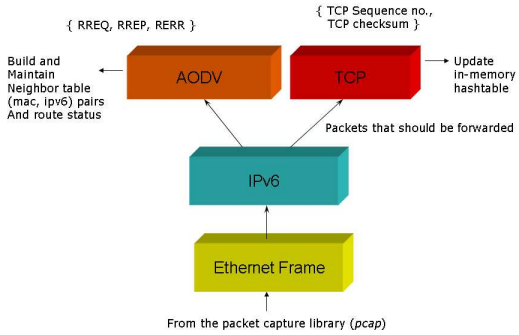

Figure 2. Packet filtering and monitoring

We use the packet capture library, `libpcap` [6, 9, 18], for capturing packets. As shown in Fig. 2 the raw packets captured by the pcap are filtered to get only IPv6 using the protocol header field in the MAC header (Ethernet in this case). Further filtering is used to separate AODV and TCP packets. We restrict ourselves to monitoring TCP data streams.

5.4.1. Building Neighbor tables The AODV control messages include special kind of RREP messages called "Hello" messages. These are used by nodes to advertise their presence and provide connectivity information in the network. These messages are broadcast

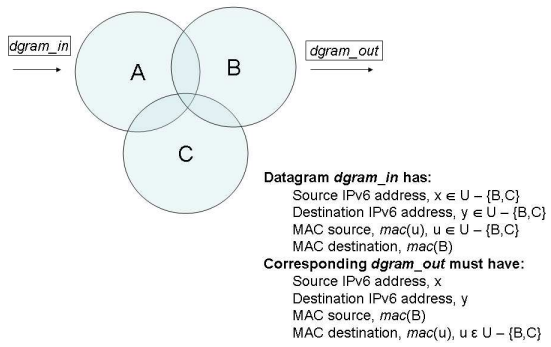

Figure 3. Monitoring traffic in radio-range

by the nodes at periodic intervals. Nodes can discover their neighbors using these messages. Also, if a neighbor moves away, the node will cease to receive it's neighbor's hello messages and thus update its routing tables. We use these messages to build neighbor tables, which consist of tuples of the form (`MAC address, IPv6 address, drop_count, route_state`), as shown in fig. 2. (`MAC address, IPv6 address`) constitute the unique key. This table is kept updated by monitoring Hello messages and RERR messages. More details on route maintenance and timeouts can be found in [2]. Data traffic of active neighbor nodes is monitored.

5.4.2. Monitoring data packets As shown in Figure 3 we monitor data packets that need to be forwarded. Referring to Figure 3, consider nodes A, B and C within radio-range of each other. Without loss of generality, let C be the monitoring node, and B be the target of monitoring. A is sending a datagram via B to some other destination. B is acting as an intermediary node forwarding packets on behalf of A. Consider the datagram *dgram_in* sent by A to B. *dgram_in* will have MAC source address of A, MAC destination address of B. But the destination IPv6 address will not be that of B, since B is not the intended recipient of *dgram_in*. Now consider the datagram that B forwards after receiving *dgram_in*. *dgram_out* will have the MAC source address of B, however the source IPv6 address in the datagram will be that of A, and not B. In fact, *dgram_in* is a datagram that B is expected to forward and *dgram_out* will be that expected datagram sent out by B, onward to its intended recipient. Packets of specific protocols can be selectively monitored using the protocol field in the IPv6 header for filtering. C being the monitoring node, will first record *dgram_in* and watch for B to transmit *dgram_out*. The processing and queuing delay at B, may vary depending on congestion and CPU load on B. Under normal circumstances, B will transmit *dgram_out* within a reasonable amount of time. If B fails to do so, then C can infer that B must have dropped the packet. When matching *dgram_in* and *dgram_out* for a particular protocol it is important to match all fields that should not be changed by B. If B mangles the packet in some malicious way, the original *dgram_in* will be unaccounted for in C's monitoring process. C will also infer such packets to have been dropped by B.

5.5. Scalability issues

For the IDS to be effective it has to be scalable. A mobile device can get overwhelmed quickly if it starts monitoring all packets in its neighborhood in promiscuous mode. A large amount of data traffic in dense networks cannot be efficiently monitored by a resource-constrained mobile device. It may be possible in certain situations to have a list of suspects that can be watched instead of all the nodes in the neighborhood. Another possibility is to monitor a random choice of neighbor nodes. Alternatively random packets can be watched to make the IDS scalable. Also the monitoring node needs to have efficient data-structures to monitor traffic efficiently in promiscuous mode.

We also have to account for the buffering capacity of nodes. Our experiments showed that during periods of congestion, or route changes, a large number of packets get buffered by intermediate nodes. Buffered packets are those that a node will watch for to be retransmitted. The mobile device is constrained in how many packets it can watch for, so a timeout is associated with each packet being watched. On a timeout, the monitoring node deems such packets to be dropped. However if these timeouts are too short, the IDS will yield a large number of false positives.

We use thresholds to distinguish between intrusions and normal behavior. Thresholds can be used to account for temporary anomalous behavior due to congestion.

5.6. Threshold-based detection

Using threshold-based detection will potentially allow a malicious node to go unnoticed if it drops a few packets intermittently. However, the potential damage caused by such intermittent packet drops will be acceptable and will not significantly affect the MANET. If a node exceeds a small threshold of such allowed "misbehavior" it will be detected and classified as intrusive. An attacker cannot significantly disrupt communication while staying under the detection-thresholds, however will be detected if the threshold is crossed.

The benefits of using thresholds are twofold. Firstly, the timeouts for packets being watched can be kept short, since most packets are expected to be retransmitted immediately. Each packet being watched accounts for memory consumed on the monitor. This means more space for newer packets and overall lower memory requirements. Secondly, false positives due to congestion are reduced. In periods of congestion, a node may queue packets to be retransmitted and not transmit them immediately, causing the monitor to assume that the packets have been dropped. Also each packet thus buffered on a neighbor node corresponds to the same packet being buffered by the monitoring node. A large number of neighbors buffering packets cause a large aggregation of such packets on the monitor itself, which occupy memory until they are timed out. Not only will they result in false positives, they have also occupied a large amount of memory before yielding possibly incorrect results.

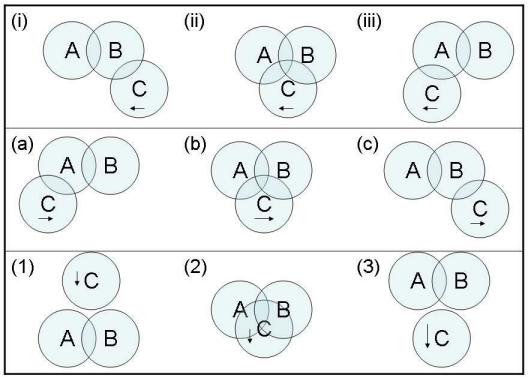

Figure 4. Effects of mobility on IDS results

Consider the three relative movements of node C with respect to A and B, B being monitored, as shown in Figure 4. The relative movement of the monitoring node with respect to its neighbors can cause false positives. In (i), (ii) and (iii) C is moving left horizontally monitoring B. When it gets out of range of B, it will continue to hear packets sent by A to B to be forwarded, but is out of range of B. Initially these will be registered as packets drops by B, however, the neighbor table will soon be updated since Hello messages from B will no longer be heard. The timeout periods are always chosen to be more than the hello message intervals, thus accounting for such situations. In (a), (b) and (c) the movement is towards B and away from A. So there will be no intrusions detected, since A will go out of range first. In (1), (2) and (3) the movement is perpendicular and equidistant from A and B. Trivially, C can hear both A and B or none, so there cannot be any false positives.

5.7. IDS validation

To test the IDS functionality, we had to setup a node that could actually drop and/or mangle packets. This was done using the Linux kernel modules `ip6table_mangle` and `ip6_queue` (userspace packet queuing using `libipq`). Perlipq [11], a Perl extension to Linux iptables for userspace queuing via `libipq` was used. The process involves adding a rule to `ip6tables` to intercept all packets to be forwarded by the node, to be queued to userspace. Perlipq then allows these packets to be manipulated by the Perl program and then passed back to the kernel. The Perl program can mangle the payload, drop the packet or return it without modifying it. We added a rule to `ip6tables` to queue all TCP packets to be forwarded, to be queued for userspace handling. Using the Perl program we configured the "malicious" node to have particular drop rates. The IDS immediately detected the dropped packets and reported them. If the drop rate exceeded the threshold value of the IDS, the IDS reported an intrusion and logged the incident. We observed that under normal traffic conditions hardly any packets are dropped by intermediate nodes when they are forwarding packets.

6. Performance Analysis

We used the `ping6` utility for sending ICMP6 echo requests to determine reachability and response times. Ping packets are given the lowest priority in packet classifying routers and are indicators of the worst quality path to the destination. We setup the iPAQs in a linear chain using `ip6tables` to drop packets from specific MAC addresses at each node, to achieve this linear chain without physically separating the iPAQs out of radio range to get such a formation. The results of the ping tests are shown in figure 5. The AODV parameters used in the tests are shown in table 2.

Referring to figure 5, the response times of `ping6` packets is shown for destinations that are 1, 2 and 3 hops away. The first column labeled AODV shows the response time of the original AODV implementation that we used to build the secure version. The second column indicates the response time of SecAODV with all its security features like signature verification turned off, but using the additional SecAODV header is shown. Finally the last column indicates the response time of SecAODV with all the security features enabled. We observe that

Parameter	Value (ms)
NODE_TRAVERSAL_TIME	100
NET_TRAVERSAL_TIME	4000
NET_DIAMETER	20
PATH_DISCOVERY_TIME	2000
HelloInterval	2000
ActiveRouteTimeout	4000
DeletePeriod	20000
RouteTimeout	8000
ReverseRouteLife	8000

Table 2. AODV parameters

1 hop	AODV	Insecure	SecAODV
Min.	1.67	2.2	2.2
Avg.	4.1	4.7	119.7
Max.	2.71	2.76	10.14
2 hops	AODV	Insecure	SecAODV
Min.	29.4	79	71.1
Avg.	37.5	169.8	205.6
Max.	31.67	123.89	145.8
3 hops	AODV	Insecure	SecAODV
Min.	-	185.8	122.4
Avg.	-	469.5	218.3
Max.	-	268.67	167.95

Figure 5. Ping6 response times using plain AODV version, SecAODV with all security features disabled, and SecAODV with all security features enabled

the packet loss is not significantly affected by the additional overheads of signature verification, during route maintenance at each node. The response times however indicate that there is delay introduced into the packet traversal time. With faster processors and larger memories the decryption and signature verification will be much faster. These results prove that SecAODV does not significantly add to the routing overhead and/or cause packet loss. We observed a large packet loss of ICMP6 packets in the original version. SecAODV however does not add to the packet loss, the packet loss remained exactly the same, though the response times increased. We note that the HUT AODV implementation [21] was tested in the AODV Interop Event [3] with only two hops. We got 100% packet loss with ping, with more than two hops using HUT AODV.

Figure 6 shows the data rates for encryption and decryption data rates using different RSA keylengths. Refer to section 4.4 for details on the signing and verification processes involved in handling AODV control messages to create and maintain routes.

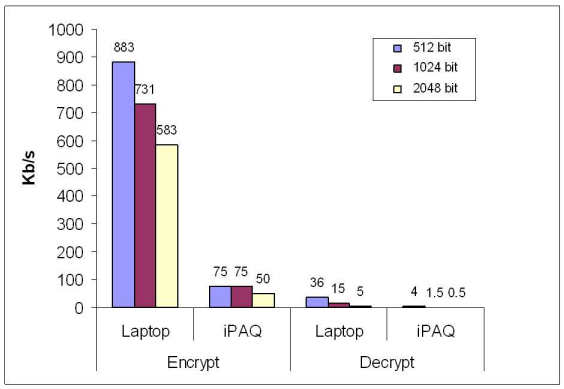

Figure 6. Data rates for encryption and decryption using RSA keys

7. Security Analysis

7.1. SecAODV security analysis

As discussed earlier, a series of routing disruption attacks have been identified in previous works. In this section we discuss how the SecAODV resists attacks by non-colluding adversaries.

"Routing disruption attacks" - in which the adversary attempts to forge a route request or a route reply by masquerading as another sender node or destination node - are prevented since either the IP verification or signature verification will fail. As long as the IP address of a node and its public key are cryptographically bound, the attacker can not successfully spoof another node's address and open or take over a communication channel unless it gains access to the victim's private key.

An attacker might also try to initiate route replies without receiving a route request. This kind of attack has minimal impact since the attacked node can easily refuse communication with a node for which it did not request a route.

Alternatively, an attacker can replay a cached route reply. This kind of attack is prevented since the protocol maintains status via sequence numbers that are signed material. As designed, the protocol drops packets that contain older (smaller) sequence numbers that the ones known to the node.

Moreover, by including the destination and originator sequence numbers in the signed material, the SecAODV

prevents "rushing attacks" in which a malicious node rushes spurious messages in which the attacker modified any of these two fields making the legitimate packet look old or as a duplicate. As long as the private keys of the end nodes are not compromised, the attacker is not capable of modifying any of these fields and thus it is not able to succeed with the planned attack.

An attacker might signal false errors thereby inducing a "resource consumption attack" or "resource exhaustion attacks" by forcing the originator node to initiate large numbers of route requests, unless the protocol is optimized to store more than one distinctive route for each destination. This solution can also minimize the impact of "grey hole" attacks in which a malicious node drops all or selective packets.

Another "resource consumption attack" is to initiate a lot of route requests, thereby causing congestion in the network. This attack can be mitigated by setting an "acceptance rate," thus limiting the number of route requests a node can accept and process per clock tick.

The SecAODV also prevents the "invisible-node attack" or "man-in-the-middle attack" in which a malicious nodes masquerades as the destination node in the communication with the originator and as the originator in the communication with the destination by enforcing IP and signature verification. Unless the malicious node possesses the private keys of both end nodes, the attacker cannot successfully play the man in the middle role.

7.2. IDS security analysis

While the use of signed control messages in a routing protocol like SecAODV can prevent routing disruption attacks, it is possible for an attacker to selectively drop only data packets. So the IDS reinforces the MANET security by detecting such grey hole attacks. The IDS is able to detect dropped and mangled packets. In the current implementation, the IDS does not distinguish between mangled packets and dropped packets, since the IDS watches for exact retransmissions. Every time a packet is faithfully retransmitted the corresponding packet is removed from the watch-list by the IDS. Mangled packets will not match any packets the IDS is watching for retransmission, and thus timeouts will cause the IDS to deem those to have been dropped. In case of TCP streams, it is possible to distinguish mangled packets from dropped packets, using the TCP sequence number and byte count. From the sequence number in the TCP packet, we can determine which part of the stream the packet belongs to and use it to determine if the intermediate node has mangled the data in any way. It is important to establish thresholds for classifying detected intrusive behavior.

8. Conclusions and Future work

In this paper we briefly described the inherent vulnerabilities of mobile devices in MANETs and several attacks possible on such devices. We presented related work in this area and presented the design and implementation of our secure routing protocol SecAODV and an IDS. The IDS is routing protocol-independent, though in this case we have used SecAODV for routing. The role of the routing protocols is just to create and maintain routes. Even after protecting the network from routing disruption attacks, packet mangling attacks and grey holes, denial of service attacks that use MAC vulnerabilities to disrupt communication are still possible. However such attacks cannot be prevented at higher networking layers, rather security mechanisms need to provided in the MAC protocol itself.

Nodes can operate on their own, however for propagating information on misbehaving nodes a platform to enable collaboration for dissemination of such IDS data is needed. The scope of a host based IDS deployed on a mobile device is limited to its radio range. We are currently implementing a collaborative IDS which will offer a collective response to misbehaving or intrusive nodes. In addition to using thresholds we are also working on using signal strengths of neighboring nodes for detecting misbehaving nodes. Potentially an IDS may assume that a neighboring node is dropping packets, when in fact, the node simply moved out of range of the monitoring node. A low signal strength will help determine the distance of the neighboring node and thus help decide if a node is misbehaving or has simply moved out of range. Also it will be helpful in selection of nodes to monitor and increase the scalability and detection accuracy of the IDS.

Acknowledgements

We would like to thank Vladimir Korolev and Soren Johnson for their technical help in this project.

References

[1] R. Bobba, L. Eschenauer, V. Gligor, and W. Arbaugh. Bootstrapping security associations for routing in mobile ad-hoc networks, May 2002.

[2] C. Perkins and E. Belding-Royer and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing, July 2003.

[3] Elizabeth M. Belding-Royer. Report on the AODV interop. http://www.cs.ucsb.edu/ ebelding/txt/interop.ps, June 2002.

[4] Y.-C. Hu, D. B. Johnson, and A. Perrig. Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*, page 3. IEEE Computer Society, 2002.

[5] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: a secure on-demand routing protocol for ad hoc networks. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 12–23. ACM Press, 2002.

[6] V. Jacobson, C. Leres, and S. McCanne. TCPDUMP group's release 3.8.3.

[7] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[8] U. Lu, B.; Pooch. Cooperative security-enforcement routing in mobile ad hoc networks. In *Mobile and Wireless Communications Network, 2002. 4th International Workshop on, Vol., Iss.*, pages 157–161, 2002.

[9] Martin Casado. Packet Capture With libpcap and other Low Level Network Tricks.

[10] G. Montenegro and C. Castelluccia. Statistically unique and cryptographically verifi able(sucv) identifi ers and addresses, 2002.

[11] J. Morris. Perlipq: Perl extension to Linux iptables userspace queueing via libipq. http://www.intercode.com.au/jmorris/perlipq/.

[12] P. Papadimitratos and Z. Haas. Secure routing for mobile ad hoc networks, January 2002.

[13] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *INFOCOM (3)*, pages 1405–1413, 1997.

[14] J. Parker, J. L. Undercoffer, J. Pinkston, and A. Joshi. On Intrusion Detection in Mobile Ad Hoc Networks. In *23rd IEEE International Performance Computing and Communications Conference – Workshop on Information Assurance*. IEEE, April 2004.

[15] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, 1994.

[16] R. Hinden and S. Deering. Internet Protocol Version 6 (IPv6) Addressing Architecture, April 2003.

[17] T. Aura. Cryptographically Generated Addresses (CGA), February 2004.

[18] Tim Carstens. Programming with pcap.

[19] C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt. A specifi cation-based intrusion detection system for aodv. In *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pages 125–134. ACM Press, 2003.

[20] Y.-C. Tseng, J.-R. Jiang, and J.-H. Lee. Secure bootstrapping and routing in an ipv6-based ad hoc network. In *ICPP Workshop on Wireless Security and Privacy*, 2003.

[21] Tuominen A. HUT AODV for IPv6 User Guide and Function Reference Guide.

[22] S. Yi, P. Naldurg, and R. Kravets. Security-aware ad hoc routing for wireless networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 299–302. ACM Press, 2001.

[23] M. G. Zapata. Secure Ad hoc On-Demand Distance Vector (SAODV) Routing. In *Internet Draft*, 2002.

[24] Y. Zhang and W. Lee. Intrusion detection in wireless ad-hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 275–283. ACM Press, 2000.