To appear in the Proceedings of the 3rd International Conference on Pervasive Computing and Communications(PerCom 2005), Kauai Island, Hawaii.

Secure Routing and Intrusion Detection in Ad Hoc Networks

Anand Patwardhan
Jim Parker
and Anupam Joshi
UMBC, Baltimore, MD 21250
{anand2, jparke2, joshi}@cs.umbc.edu

Michaela Iorga
and Tom Karygiannis
NIST, Gaithersburg, MD 20899
{miorga, karygiannis}@nist.gov

***Abstract*— Numerous schemes have been proposed for secure routing and Intrusion Detection for ad hoc networks. Yet, little work exists in actually implementing such schemes on small handheld devices. 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 System (IDS) for ad hoc networks. Security features in the routing protocol include mechanisms for non-repudiation and authentication, 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.**

I. 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, GPRS etc. Due to technology limitations, however, wireless access to the service providing infrastructure (cell towers, WLAN base-stations) is limited to particular areas. Moreover, buildings and other physical obstructions further restrict availability.

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 CA or a KDC 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 [19], DSR [11], AODV [18], TORA [16] etc. A majority of these protocols assume 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 IDSs 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

This research was supported by NSF award 9875433, and a grant from NIST

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 and are discussed in the related work section. Simulations and illustrations have been used to validate the feasibility of proposed schemes for secure routing and intrusion detection. We propose a combination of a secure routing protocol and an IDS for strengthening the defense of a MANET. To the best of our knowledge, this IDS is the first actual implementation deployed on handheld devices. The IDS is based on an algorithm proposed in our previous work [17]. We also describe the implementation of our secure routing protocol, SecAODV. We present a detailed analysis of issues involved in the implementation and deployment of a secure routing protocol and 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, and are not apparent from simulations.

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.

II. Background and Related Work

A. Secure Routing Protocols

As noted earlier, a majority of the proposed routing protocols assume non-hostile environments, where nodes faithfully forward packets, and malicious nodes are absent. 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 [25], Ariadne [8], SEAD [7], CSER [12], SRP [15], SAAR [24], BSAR [3], and SBRP [22].

Our implementation of the SecAODV is similar to the protocol proposed in BSAR [3] and SBRP [22] for DSR. SecAODV is a highly adaptive distributed algorithm designed for IPv6-based MANETs that does not require: (1) prior trust relations between pairs of nodes (e.g. a trusted third party or a distributed trust establishment), (2) time synchronization between nodes, or (3) 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 [3], [13] 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) [1]. SUCVs associate a host's IPv6 address with its public key that provides verifiable proof of ownership of that IPv6 address to other nodes.

B. 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.

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 their observable radio transmission range.

Zhang and Lee [26] 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 [20] and specification based algorithms [21]. Tseng *et al.* [21] 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.

III. SecAODV implementation details

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 I

iPAQ 3800 Series Specifications

A. Assumptions and Observations

We assume that interfaces have a promiscuous mode to monitor traffic of neighboring nodes. Key lengths are chosen to be 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. It is also assumed that normal packet drop rates can be dynamically determined and thresholds established to distinguish malicious behavior from trustworthy conduct. We do not require the MAC addresses to be unforgeable, since the SUCV identifiers provide secure binding between IPv6 addresses and public keys. Identity is not determined by the MAC address alone. Address spoofing can be detected since signature verification will fail unless private keys have been compromised. Also, since misbehavior is associated with the IPv6 address, the attacker may periodically change his/her IPv6 address, but at the additional expense of computing a SUCV every time. Consequently such an attack is largely ineffective, and quite expensive for the attacker.

B. SecAODV

1) Overview: The SecAODV implements two concepts which are common features in both BSAR [3] and SBRP [22]:

- Secure binding between 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. The address auto-configuration feature available in IPv6 that allows IP auto-configuration for the nodes on a need basis, is of special importance.

The SecAODV implementation follows Tuominen's design [23] which uses two kernel modules `ip6_queue`, `ip6_nf_aodv`, and a userspace daemon `aodvd`.

2) Secure Address Auto-Configuration and Verification: To join a MANET, a node executes a script that sets its Service Set Identifier (SSID), 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 are then used for the IPv6 address because 2 bits of the address space are reserved. The final IPv6 address 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) [6]. A source node uses the secure binding to authenticate its IPv6 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 to make its presence known.

C. Working of SecAODV

The AODV protocol [18] is comprised of two basic mechanisms, viz., *route discovery* and *maintenance of local connectivity*. The SecAODV protocol adds security features to the basic AODV mechanisms, but is otherwise identical. A source node S 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 SecAODV `RREQ` message differs from the one proposed in [18], it additionally contains the RSA public key of the source node S and is digitally signed to ensure authenticity and integrity of the message (refer to Fig. 1). Upon receiving a RREQ message, each node authenticates the source S, by verifying the message integrity (see section $III.B$) , and by verifying the signature against the provided public key. Upon successful verification, the node updates its routing table with S's address and the forwarding node's address. If the message is not addressed to it, it rebroadcasts the RREQ. When D receives the RREQ, it constructs a signed route reply message (RREP) addressed to the source node S, which includes the D's public key, as shown in Fig. 1. D then unicasts the RREP back to the neighboring node from which the RREQ was received. 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 for D and routes it towards S. If a route entry for S does not exist or has expired, the message is dropped and an error message is sent to all affected neighbors. If S does not receive any reply in a predetermined amount of time, it rebroadcasts new route requests. *Maintenance of local connectivity* mechanism is optionally achieved by periodically broadcasting *Hello*-messages. In our implementation these messages are signed and contain the sender's public key for authentication and message integrity verification.

IV. Intrusion Detection

Although encryption and signed headers are intrusion prevention measures, vulnerabilities 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. The use of SUCVs is thus well-suited for such situations.

The effectiveness of a collaborative IDS depends on the amount of data that can be collected individually. Longer presence increases the availability of meaningful data. However the degree of mobility has a significant impact on the effectiveness of the IDS. Routing errors and packet drops due to increased mobility may mask malicious behavior, however malicious nodes cannot significantly affect routing either.

A. Design Goals

1) Scalability: The effectiveness of the IDS will depend on its scalability. Snooping on all packet traffic is prohibitively expensive for most resource-constrained mobile devices, especially when number of nodes within radio-range increase. Dense networks or larger radio-ranges of new wireless technologies will have a large number of neighbor nodes.

2) Platform for a collaborative IDS: Individual nodes with IDS deployments can only monitor within their radio-range. It is necessary to aggregate such data to detect anomalies and malicious colluding activity in the network through peer interactions. The IDS should enable collection of local audit data.

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.

B. 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. Local audit data can then be aggregated by some centralized/distributed algorithm, to detect ongoing attacks. 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.

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

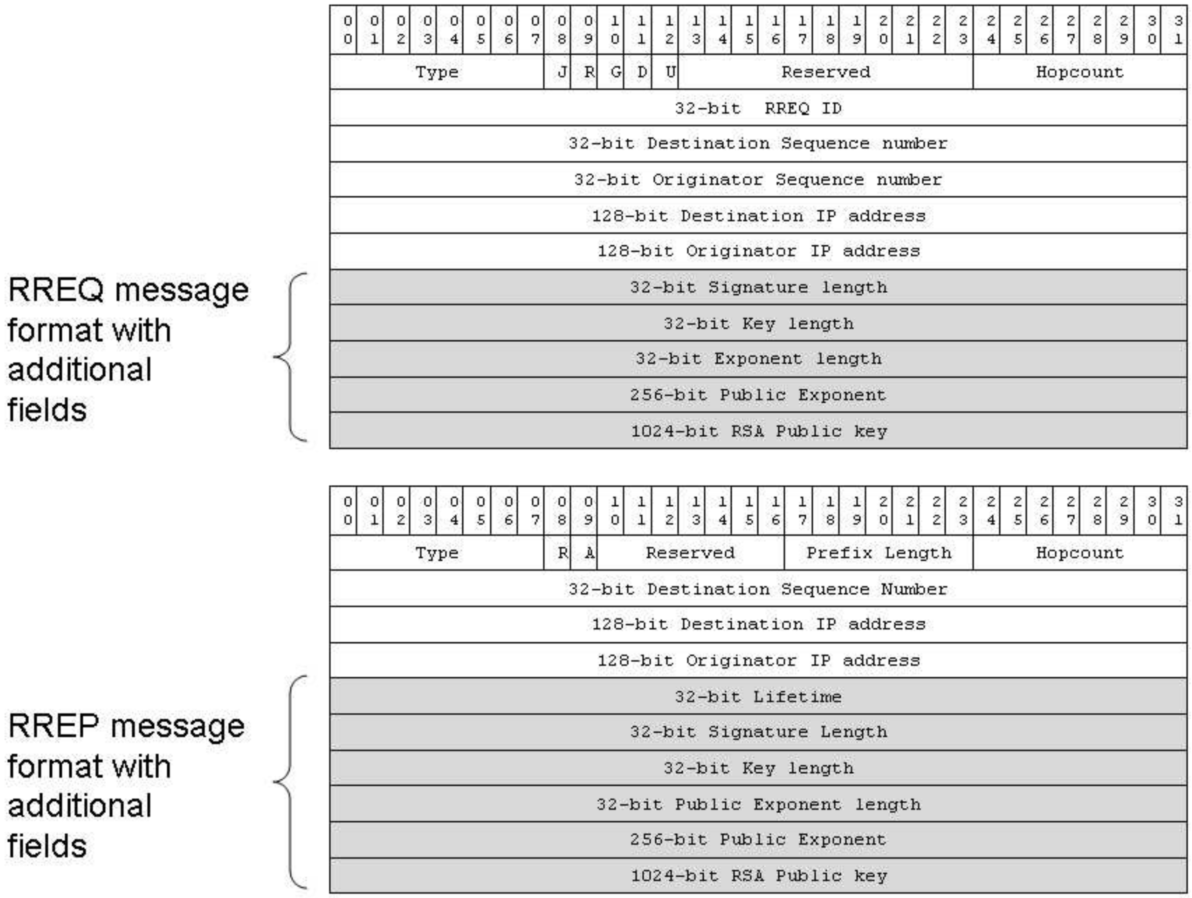

Fig. 1. SecAODV message formats

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 detected in the signature verification at the receiver.

C. *Stateful packet monitoring*

We use the packet capture library, `libpcap` [4], [5], [10], for capturing packets. As shown in Fig. 2 the captured raw packets are filtered to get only IPv6 using the protocol header field in the MAC header. Further filtering is used to separate AODV and TCP packets. We restrict ourselves to monitoring TCP data streams.

1) Building Neighbor tables: The AODV control messages include special kind of `RREP` messages called "Hello" messages. These messages are broadcast by the nodes at periodic intervals. Nodes can discover their neighbors using these messages.

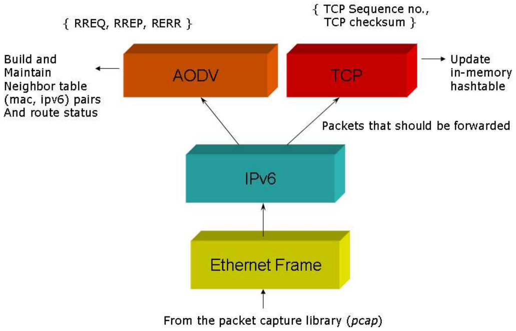
{ RREQ, RREP, RERR }
{ TCP Sequence no., TCP checksum }
Build and Maintain Neighbor table (mac, ipv6) pairs And route status
AODV
TCP
Update in-memory hashtable
Packets that should be forwarded
IPv6
Ethernet Frame
From the packet capture library (*pcap*)

Fig. 2. Packet filtering and monitoring

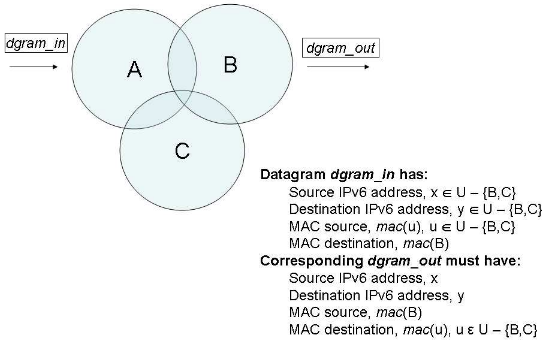
dgram_in
A
B
dgram_out
C
Datagram dgram_in has:
Source IPv6 address, x ∈ U – {B,C}
Destination IPv6 address, y ∈ U – {B,C}
MAC source, mac(u), u ∈ U – {B,C}
MAC destination, mac(B)
Corresponding dgram_out must have:
Source IPv6 address, x
Destination IPv6 address, y
MAC source, mac(B)
MAC destination, mac(u), u ε U – {B,C}

Fig. 3. Monitoring traffic in radio-range

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.

2) Monitoring data packets: As shown in Fig. 3 we monitor data packets that need to be forwarded. Referring to Fig. 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. Another possibility is that B mangles 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 maliciously mangles the packet, the original *dgram_in* will not match any *dgram_out*. C detects mangling by looking at the TCP sequence number, checksum and byte count.

D. 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.

E. 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. Thresholds allow for short timeouts, for packets being watched, 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.

F. IDS validation

To test the IDS functionality, we setup a node that could drop and/or mangle packets. This was done using the Linux kernel modules `ip6table_mangle` and `ip6_queue` (userspace packet queuing using `libipq`). Perlipq [14], 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. 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.

V. Security Analysis

A. SecAODV security analysis

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 IPv6 address verification or signature verification will fail. As long as the IPv6 address of a node and its public key are cryptographically bound, the attacker can not successfully spoof another node’s address unless it victim’s private key is compromised.

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 ignore packets from a node to 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 contained in the signed header. As designed, the protocol drops packets that contain sequence numbers older than those currently known. Moreover, by including the destination and originator sequence numbers in the signed material, the SecAODV prevents “rushing attacks” [9] 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 immune to rushing attacks.

One kind of “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.

SecAODV also prevents the “man-in-the-middle attack” by enforcing IP and signature verification. Unless the malicious node possesses the private keys of both end nodes, the attacker cannot launch a “man-in-the-middle” attack.

B. 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.

VI. Performance Analysis

We used the `ping6` utility for sending ICMP6 echo requests to determine reachability and response times. 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 Fig. 4. The AODV parameters used in the tests are shown in table II.

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 II

AODV PARAMETERS

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

Fig. 4. Ping6 response times in seconds using plain AODV version, SecAODV with all security features disabled, and SecAODV with all security features enabled

Referring to Fig. 4, the response times of `ping6` packets are 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 the packet loss is not significantly affected by the additional overhead of signature verification during route maintenance at each node. The response times however indicate that there is delay introduced in 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 [23] was tested in the AODV Interop Event [2] with only two hops. We got 100% packet loss with `ping6`, with more than two hops using HUT AODV.

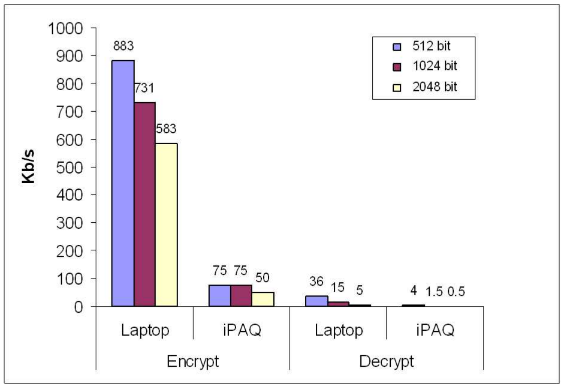

Fig. 5. Data rates for encryption and decryption using RSA keys

Fig. 5 shows the data rates for encryption and decryption data rates using different RSA keylengths.

VII. 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 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.

VIII. Acknowledgments

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

References

[1] T. Aura. Internet Draft: Cryptographically Generated Addresses (CGA). http://www.ietf.org/proceedings/04mar/I-D/draft-ietf-send-cga-05.txt, February 2004.

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

[3] R. Bobba, L. Eschenauer, V. Gligor, and W. Arbaugh. Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks, May 2002.

[4] T. Carstens. Programming with pcap. http://www.tcpdump.org/pcap.htm.

[5] M. Casado. Packet Capture With libpcap and other Low Level Network Tricks. http://www.cet.nau.edu/˜mc8/Socket/Tutorials/section1.html.

[6] R. Hinden and S. Deering. RFC 3513: Internet Protocol Version 6 (IPv6) Addressing Architecture, April 2003.

[7] 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.

[8] 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.

[9] Y.-C. Hu, A. Perrig, and D. B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *Proceedings of the 2003 ACM workshop on Wireless security*, pages 30–40. ACM Press, 2003.

[10] V. Jacobson, C. Leres, and S. McCanne. TCPDUMP group's release 3.8.3. http://www.tcpdump.org/.

[11] 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.

[12] B. Lu and U. 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.

[13] G. Montenegro and C. Castelluccia. Statistically Unique and Cryptographically Verifiable(SUCV) identifiers and addresses. citeseer.ist.psu.edu/montenegro02statistically.html, 2002.

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

[15] P. Papadimitratos and Z. Haas. Secure Routing for Mobile Ad Hoc Networks. In *Communication Networks and Distributed Systems Modeling and Simulation Conference*, pages 27–31, January 2002.

[16] V. D. Park and M. S. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *INFOCOM (3)*, pages 1405–1413, 1997.

[17] 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.

[18] C. Perkins, E. Belding-Royer, and S. Das. RFC 3561: Ad hoc On-Demand Distance Vector (AODV) Routing, July 2003.

[19] 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.

[20] B. Sun, K. Wu, and U. W. Pooch. Alert aggregation in mobile ad hoc networks. In *Proceedings of the 2003 ACM workshop on Wireless security*, pages 69–78. ACM Press, 2003.

[21] C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt. A specification-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.

[22] 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.

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

[24] 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.

[25] M. G. Zapata. Internet Draft: Secure Ad hoc On-Demand Distance Vector (SAODV) Routing. http://www.cs.ucsb.edu/˜ebelding/txt/saodv.txt, 2002.

[26] 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.