Experimental Comparisons between SAODV and AODV Routing Protocols

Yuxia Lin, A. Hamed Mohsenian Rad, Vincent W.S. Wong,
Department of Electrical and Computer Engineering
The University of British Columbia, Vancouver, BC, Canada
e-mail: {yuxial, hamed, vincentw}@ece.ubc.ca

Joo-Han Song
4G System Laboratory
Samsung Electronics, Korea
joohan.song@samsung.com

ABSTRACT

There have been various secure routing protocols proposed for mobile ad hoc networks. Most of these protocols are analyzed by two standard techniques: simulation and security analysis. There has been a lack of work related to the performance of secure routing protocols in real network testbed. In this paper, we present quantitative results for the performance comparisons between AODV and SAODV routing protocols by using a small-scale experimental testbed, which consists of 10 laptops within a 250 m by 100 m rugby field. Apart from outdoor testing, we also compare the results with those obtained via simulation and indoor emulation. The workload includes both UDP and TCP traffic. Results show that SAODV is effective in preventing routing message tampering and data dropping attacks. For outdoor experiments, we also estimate the average distance within a communication gray zone under different bit rates.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols − *Routing Protocols*

General Terms

Security, experimentation

Keywords

Wireless ad hoc networks, security, routing, testbed

1. INTRODUCTION

A Mobile Ad-hoc NETwork (MANET) consists of a set of wireless mobile nodes communicating with each other without any centralized control or fixed network infrastructure. Over the past decade, many routing protocols have been proposed in the literature (e.g., [1]-[4]). Examples include AODV (Ad hoc On-demand Distance Vector)[1] and DSR (Dynamic Source Routing) [2] protocols. These protocols have been studied extensively. There have been various testbed experiments for ad hoc networks (e.g., DSR [17], ABR [18], link-level measurements [19]). Each one differs by the routing protocol being tested, the number of nodes, and the implementation details.

For the implementation study of the AODV routing protocol, two of them which are RFC-compliant and have passed the interoperability testing include AODV-UU [20] and Kernel-AODV [22]. Others include AODV-UIUC [23] and AODV-Windows from UCSB [24] and UoBWinAODV [25]. In [29], Chin *et al*. compared AODV and DSDV (Destination-Sequenced Distance Vector) in a testbed with 5 nodes. In [30], Gray *et al.* compared the packet delivery ratio between AODV, APRL (Any Path Routing without Loops), ODMRP (On-Demand Multicast Routing Protocol), and STARA (Traffic-dependent Adaptive Routing Algorithm) in both outdoor and indoor environments. The outdoor experiments consisted of 33 nodes moving within an athletic field. Some of the recent work on ad-hoc network testbed included the study of directional antennas [32] as well as multi-channel wireless mesh networks [33][34].

Most of these ad-hoc routing protocols assume that there is an implicit trust-your-neighbor relationship in which all the neighboring nodes behave properly. However, real MANETs are subject to attacks by malicious users, who try to paralyze the network by manipulating the messages (e.g., dropping all data or control packets, sending incorrect route advertisement messages). This problem is further complicated by the lack of centralized management control, error-prone wireless channels, and the dynamic changes in network topology due to node mobility.

A number of secure ad-hoc routing protocols have been proposed [5]-[11] with the aim of preventing different types of attacks (e.g., message tampering, message dropping, message replay). Most of these protocols are analyzed by using two techniques: simulation and security analysis (e.g., [13][14][16]). The implementation of ARAN [8] is described in [31]. Although there is some work on security in wireless sensor networks, the authors are not able to find other experimental works on secure routing in mobile ad hoc networks in the literature.

Our work is motivated by the following questions:

1) Do user mobility and physical layer's impairments (e.g., multipath fading, interference) affect the effectiveness of secure routing protocols?

2) Does the extra control overhead and processing incurred by the security mechanisms affect the performance in the upper layer (e.g., TCP, UDP)?

To answer the above questions, we set up an experimental testbed and evaluate the performance between AODV [1] and SAODV (Secure AODV) [5] routing protocols under different environments. The outdoor testbed consists of ten 802.11-enabled laptops within a 250 m by 100 m rugby field. Although this is a small testbed, we did learn some lessons from the outdoor test experiments. In addition, we compare the results with those

WMuNeP'05, October 13, 2005, Montreal, Quebec, Canada.
Copyright 2005 ACM 1-59593-183-X/05/0010...$5.00.

obtained via simulation and mobility emulation in indoor environments. We chose AODV in our study because it is one of the ad hoc routing protocols standardized within the IETF (Internet Engineering Task Force).

The main contributions of this paper are as follows:

(1) By extending the AODV-UU module [20], we successfully implemented the security features (i.e., digital signature, hash chains) necessary for SAODV in linux environment.

(2) Three sets of results (simulation, indoor emulation, outdoor) under TCP or UDP traffic confirm that SAODV is effective in preventing the control message tampering and data dropping attacks.

(3) Measurements from outdoor experimental testing show that the width of the communication gray zone is about 9 m and 18 m under the transmission rate of 2 Mb/s and 11 Mb/s, respectively.

This paper is organized as follows. Section 2 provides an overview of AODV and SAODV. In Section 3, we describe the settings for the simulation, indoor, and outdoor experiments. Results and discussions are presented in Section 4. Conclusions are given in Section 5.

2. ON-DEMAND ROUTING PROTOCOLS

In this section, we first provide an overview of AODV [1] and identify several potential security issues. We then describe SAODV [5], which can protect the routing messages in AODV.

2.1 AODV

The AODV routing protocol [1] is an on-demand variation of the distance vector routing protocol. When a source node desires to send a message to a certain destination node to which it does not have a valid route, it initiates a route discovery process. The source node broadcasts an RREQ (Route REQuest) message to its neighbors, which then forward the request to their neighbors, and so on, until either the destination or an intermediate node with a route to the destination in its routing table is reached. During the process of forwarding the RREQ, an intermediate node record in its routing table (i.e., precursor list) the address of the neighbor from which the first copy of the broadcast packet is received, thereby establishing a reverse path. Additional copies of the same RREQ received later are discarded. Once the RREQ reaches the destination or an intermediate node with a route, the respective node responds by unicasting an RREP (Route REPly) message back to the neighbor from which it first received the RREQ, which relays the RREP backward via the precursor nodes to the source node.

Routes are maintained as follows: HELLO beacons are sent periodically via broadcast to the neighboring nodes. When a source node moves, it has to re-initiate the route discovery protocol to find a new route to the destination. On the other hand, when an intermediate node along the route moves, its upstream neighbor will notice route breakage due to the movement and propagate an RERR (Route ERRor) message to each of its active upstream neighbors. These nodes in turn propagate the RERR packet to their upstream neighbors, and so on until the source node is reached. The source node may then choose to re-initiate the route discovery for that destination if a route is still desired.

2.2 Security Issues in AODV

In this section, we analyze the security threats and describe the requirements for AODV routing protocol to mitigate these threats. A node is *malicious* if it is an attacker that cannot authenticate itself as a legitimate node due to the lack of valid cryptographic information. A node is *compromised* if it is an inside attacker who is behaving maliciously but can be authenticated by the network as a legitimate node and is being trusted by other nodes. A node is *selfish* when it tends to deny providing services for the benefit of other nodes in order to save its own resources. Several attacks can be launched against the AODV routing protocol:

Message tampering attack: An attacker can alter the content of routing messages and forward them with falsified information. For example, by reducing the hop-count field in either an RREQ or RREP packet, an attacker can increase its chance to be an intermediate node of the route. A selfish node can relieve the burden of forwarding messages for others by setting the hop-count field of the RREQ to infinity. Simulation results in [12] show that a single attacker can drop up to 75% of packets by manipulating destination sequence numbers in some scenarios.

Message dropping attack: Both attackers and selfish nodes can intentionally drop some (or all) routing and data messages. Since all the mobile nodes within a MANET function as both end hosts and routers, this attack can paralyze the network completely as the number of message dropping increases.

Message replay (or wormhole) attack: Attackers can re-transmit eavesdropped messages again later in a different place. One type of replay attacks is the wormhole attack. A wormhole attacker can tunnel an RREQ directly to a destination node. Since a wormhole attacker may not increase the hop-count field value, it prevents any other routes from being discovered. The wormhole attack can be combined with the message dropping attack to prevent the destination node from receiving packets.

The security requirements for AODV routing protocol include:

(1) Source authentication: The receiver should be able to confirm that the identity of the source is indeed who or what it claims to be.

(2) Neighbor authentication: The receiver should be able to confirm that the identity of the sender (i.e., one hop previous node) is indeed who or what it claims to be.

(3) Message integrity: The receiver should be able to verify that the content of a message has not been altered either maliciously or accidentally in transit.

(4) Access control: It is necessary to ensure that mobile nodes seeking to gain access to the network have the appropriate access rights.

There are a number of secure protocols proposed especially for AODV. Examples include SAODV (Secure AODV) [5] and ARAN (Authenticated Routing for Ad hoc Networks) [8]. An overview of SAODV is given in the next section.

2.3 SAODV (Secure AODV)

The SAODV routing protocol proposed in [5] is used to protect the routing messages of the original AODV. SAODV uses digital signatures to authenticate non-mutable fields and hash chains to authenticate the hop-count field in both RREQ and RREP messages. We now explain the operation of the hash chains.

During the route discovery process, the source node first selects a random *seed* number and sets the Maximum Hop-count (*MHC*) value. By using a hash function *h*, the source computes the *hash* value as *h(seed*) and *Top_Hash* as h^{MHC}(*seed*).

When an intermediate node receives an RREQ message, it checks whether the value of *Top_Hash* is equal to $h^{MHC-Hop\_Count}$(*Hash*). If so, it will assume that the hop count has not been altered. Before rebroadcasting the RREQ to the neighboring nodes, the intermediate node will increment the hop-count field by one in the RREQ header and also compute the new *Hash* value by hashing the old value (i.e., *h*(*Hash*)).

Except for the hop-count field and $h^{\text{hop-count}}$(*seed*), all other fields of the RREQ are non-mutable and therefore can be authenticated by verifying the signature in the RREQ. When the destination node receives an RREQ, it generates an RREP in the same way. SAODV can also allow an intermediate node to generate an RREP by using double signature extension.

3. EXPERIMENTAL SETUP

In this section, we first describe the hardware platform and the approach we used to implement SAODV. We then describe the setup for simulation, indoor emulation, and outdoor experiments.

3.1 Hardware Platform

The testbed consists of 10 IBM Model T42 laptops. Each laptop has an Intel Pentium M 1.5 GHz CPU with 1024 KB cache, 40 GB disk space, and 512 MB of main memory. Each laptop is equipped with an IBM 11a/b/g Wireless LAN mini PCI adapter and runs on Linux kernel version 2.4.20. The IEEE 802.11b interface is used. Except for setting the ad-hoc mode and selecting the frequency band and channel number, we use the default configuration for the radio interface. In all three experiments (simulation, indoor emulation, outdoor), the auto-rate selection and RTS/CTS are disabled.

3.2 Software Infrastructure

For AODV, we use the AODV-UU implementation [20]. AODV-UU is RFC 3561 compliant and uses the Netfilter framework in Linux to run as a user space daemon. One kernel module (kaodv) is used for registering packet handling with Netfilter hooks and for modifying kernel routing table.

We are not aware of any publicly available implementation of SAODV [5]. To this end, we modified some modules in AODV-UU for SAODV. We included the hash chain functionalities for hop-count verification in the RREQ, RREP message handling modules. The MD5 [26] was used as the hash function. For the purpose of protecting routing messages with digital signatures, we ported part of the code from the ARAN (Authenticated Routing for Ad-Hoc Networks) implementation [31]. ARAN uses the OpenSSL library for certification. The non-mutable fields in the routing messages are protected by the digital signatures.

The original AODV-UU allowed intermediate nodes to send back RREP messages. This complicates the digital signature signing process, due to the difficulties to verify the authenticity of this kind of RREPs. For SAODV, we disabled the intermediate nodes' capability of sending RREPs. Only the route destination node will send a signed RREP message.

For the attacker model, we used the control message tampering and data message dropping attacks described in Section 2.2. We included an attacker module by modifying the original AODV-UU code. A routing module can be compiled as an attacker with a flag in the defs.h header file turned on. When an attacker receives an RREQ message, it will send an RREP with hop-count value equals zero. If the attacker is chosen as an intermediate relaying node, it will subsequently drop all received data packets if any nodes choose the attacker as the intermediate relaying node.

For all the modifications we made to the original AODV-UU to support SAODV, we plan to make the source codes available for public download in near future.

3.3 Parameters Used in Experiments

In all three experiments (simulation, indoor emulation, and outdoor), the network topology consists of 10 nodes. Initially, the nodes are placed randomly in a 250 m by 100 m grid. The random waypoint mobility model is used. In both simulation and indoor emulation tests, the maximum node's speed is 2 m/sec and the pause time value is 40 sec. In the outdoor experiment, each node moves with a speed of 1 m/sec and the pause time value is 0 sec.

In each test run, 3 source and destination pairs are randomly selected among the 10 nodes. All three sessions (or flows) are either UDP or TCP traffic. For UDP traffic, three Constant Bit Rate (CBR) sessions generate UDP packets from nodes 2, 4 and 6 to nodes 3, 5 and 7, respectively. The UDP packet size is 512 bytes and the CBR transmission rate is 4 packets/sec. For TCP traffic, the same 3 sources generate File Transfer Protocol (FTP) packets to the same destinations. The TCP packet size is 1000 bytes, the maximum congestion window size is 11 packets and the TCP Reno version is used.

In both simulation and indoor emulation experiments, the bit rate for the 802.11b MAC is 2 Mb/s. In the outdoor experiment, the bit rate for the 802.11b MAC is 11 Mb/s. For both AODV and SAODV, HELLO packets are sent every 1 second. A link between two nodes is declared to be broken if a HELLO packet is not received within 2 seconds. For SAODV, the size of the additional fields for RREQ, RREP, and RERR packets are 448 bytes, 448 bytes, and 404 bytes, respectively. Note that for the 448 bytes in both RREQ and RREP include signature (64 bytes), top hash (16 bytes), hash (16 bytes) certificate (339 bytes) and other header information (13 bytes). The 404 bytes in RERR include the signature (64 bytes) certificate (339 bytes) and other header information (1 byte).

3.4 Simulation Experiments

The ns-2 [27] is used for the simulation experiments. The simulation time for each test is 1800 seconds. Apart from the parameters described in Section 3.3, the transmission range of each node is 100 m and the free space model is used as the radio propagation model. The SAODV module is implemented by modifying the original AODV source code. The attacker node's behavior is also added to the source codes. During each simulation run, besides the performance comparison metrics, the instantaneous position of each node is logged to emulate the mobility pattern later used for the indoor emulation tests.

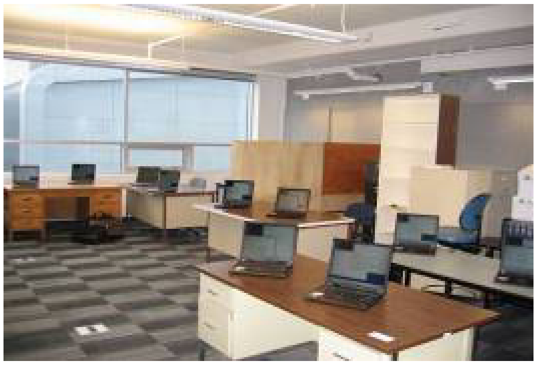


Figure 1. Indoor Testing

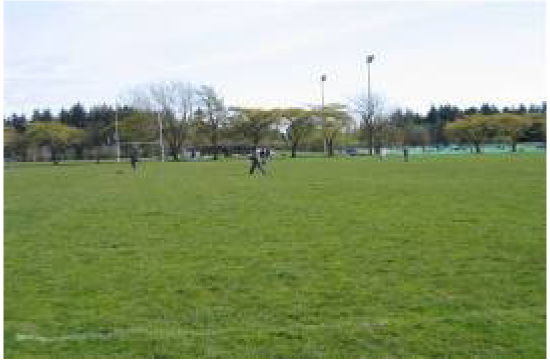


Figure 2. Outdoor Testing

3.5 Indoor Emulation Experiments

The current commercial 802.11 wireless cards have a transmission range between 100 m – 500 m. An outdoor mobile ad-hoc network testing for routing protocols requires a large coverage area, an adequate number of mobile devices and personnel for participation. This makes real field testing especially difficult. To this end, we implemented a mobility emulator MacSim which is similar to the MacKill program used in the APE project [21]. Unlike the MacKill in APE which runs as a kernel module for packet killing, we utilized the well developed packet filtering program "iptables" in Linux for filtering packets based on source MAC address to emulate the link breakage.

The MacSim program runs independently in each laptop. It synchronizes all 10 laptops using the Network Time Protocol (NTP) at the beginning of the emulation. Then, the program on each laptop reads a mobility scenario file which mandates this laptop's connectivity to all other laptops at every second's interval to emulate the laptops' movements. The status for the links among the 10 laptops is calculated from the (*x*,*y*) position trace files from ns-2 simulation. A fixed transmission range of 100 m is assumed.

During the emulation, all 10 laptops are placed in the same room (see Figure 1). The mobility trace files are generated via a random mobility model in ns-2. This facilitates the comparison of ns-2 simulation and indoor emulation results. The advantages of this emulation approach are that it facilitates the program debugging. Also, the protocol performance and different mobility scenarios can be tested and repeated in a well controlled manner. However, our current MacSim program can only simulate an ON/OFF binary state of the wireless link which may not be realistic in a wireless environment. Further improvement can be made by including some packet dropping probabilities in the MacSim program such that it has a closer resemblance to real wireless links. In addition, more realistic results can be obtained if the mobility trace file is obtained via actual outdoor testing.

In indoor emulation, all the laptops use CSMA/CA for channel access. Effectively, they are all in the same collision domain. This reduces the possible network capacity compared to a real multi-hop network. However, the carrier sense range in ns-2 simulation and real wireless cards are usually more than twice the transmission range. Given the field size we use for simulation and outdoor testing with a diameter of less than 250 meters, this puts most of the laptops within one carrier sense range as well. As a result, the approximation of one single carrier sense domain in the emulation is not significant.

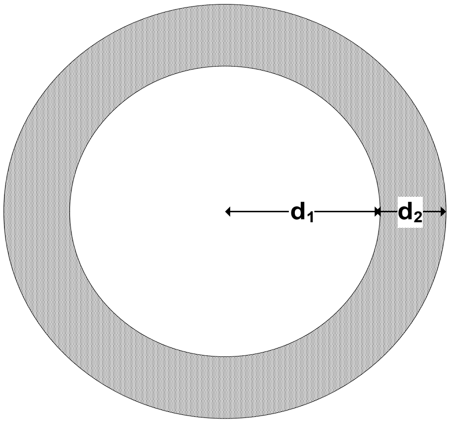
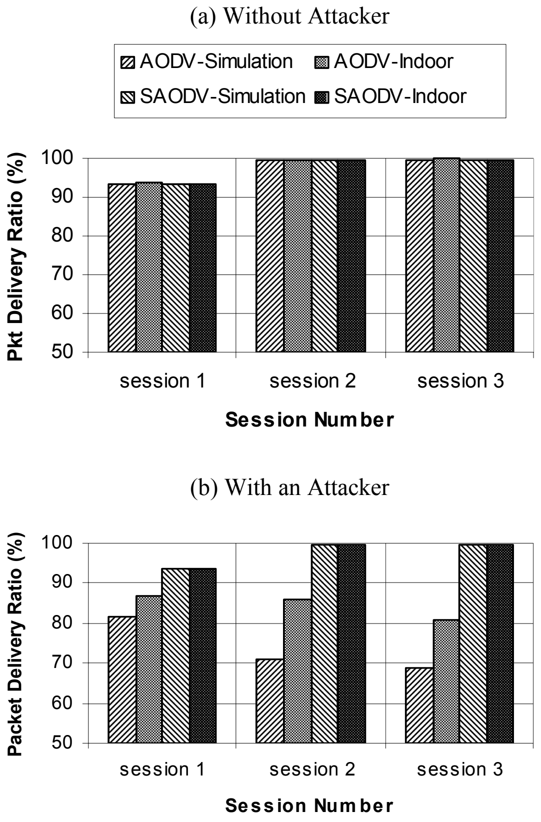


Figure 3. Communication Gray Zone

The traffic in the network is generated by using the Iperf program which supports both TCP and UDP traffic flows. During the indoor emulation, all the wireless cards were set to work in the 802.11b ad-hoc mode with the data-rate of 2 Mbps and long preamble only. This setting is used to match the corresponding ns-2 simulation parameters. To reduce possible interference from other WLANs within the university campus, we used channel #11 which is the farthest from the most commonly used channel #6.

3.6 Outdoor Experiments

The outdoor experiments were conducted in a rugby field which is near the university campus. The area consists of 1/4 sparsely clustered 3-story buildings and 3/4 open air field. We used the satellite images to confirm that the size of field is approximately 250 m by 100 m. Part of the outdoor field is shown in Figure 2. In the outdoor test, each participant held a laptop and walked randomly in the field with a speed of 1 m/s. Each test run took 6 minutes. The wireless cards were set in 802.11b ad-hoc mode with channel #11. The data rate was 11 Mb/s with auto-rate function disabled. Due to the field size constraint, the device driver was set to work in the minimum transmission power mode so that the transmission range is about 100 m.



(a) Without Attacker

(b) With an Attacker

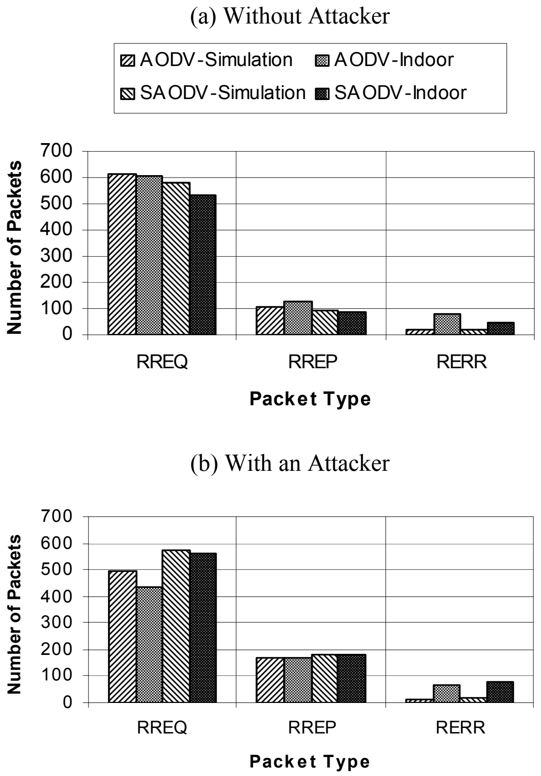
Figure 4. UDP Packet Delivery Ratio

Previous work by Lundgren *et al.* [28] has indicated the potential *communication gray zones* problem in AODV. Due to the different transmission rates and sizes of data and control packets, there may exist certain zones such that a node can only receive control packets successfully but not data packets. This implies that a node can have a valid route in its routing table, but no data packets can be sent to the next hop.

To study the effects of the communication gray zone problem, we made some measurements during the outdoor test. For this particular test, only two nodes are used. One node is the source and the other is the destination. The source node sends packets at a rate of 4 packets/sec with packet size equal to 512 bytes. The destination node moves away from the source node and measures the packet loss rate periodically as a function of the distance. As a first level approximation, there are two regions (see Figure 3). The centre of the circle is the location of the source node. When the destination node is within distance d_1, the packet delivery ratio is above 90%. When the destination node is within the gray zone (i.e., within d_2), both data and control packets begin to drop occasionally. When the destination is beyond the distance $d_1 + d_2$, both data and control packets cannot be successfully received.

4. RESULTS AND DISCUSSIONS

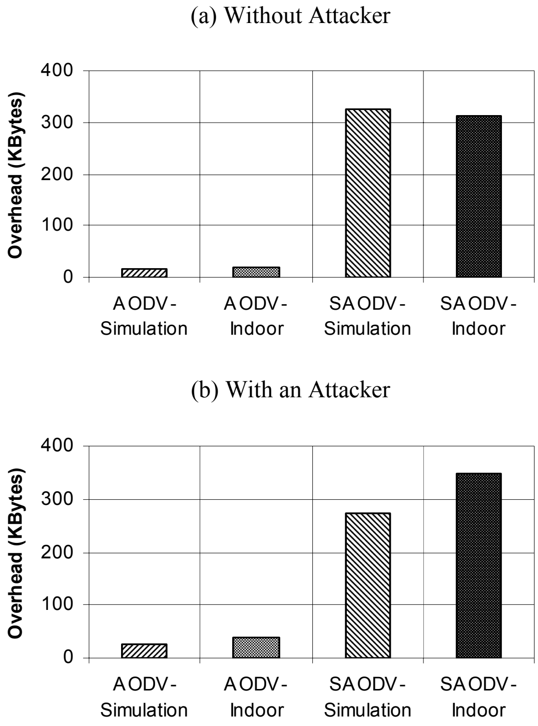
The following performance metrics are used for comparisons. For UDP traffic, the *packet delivery fraction* is defined as the



(a) Without Attacker

(b) With an Attacker

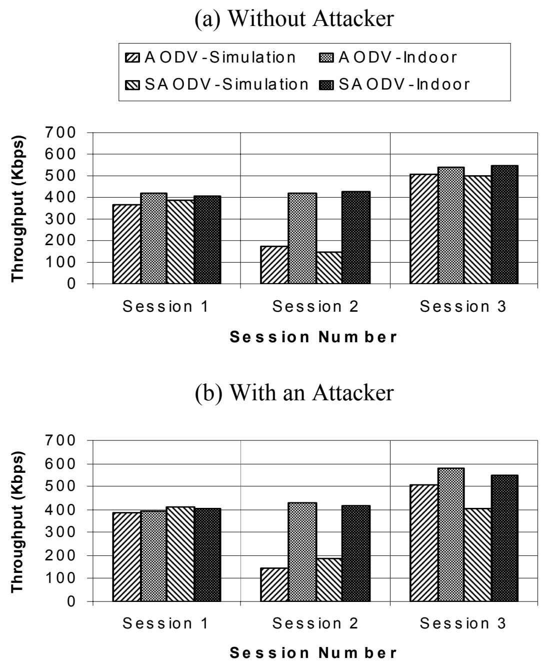
Figure 5. Routing Control Overhead (in packets)



(a) Without Attacker

(b) With an Attacker

Figure 6. Routing Control Overhead (in bytes) for UDP


(a) Without Attacker

(b) With an Attacker

Figure 7. Average TCP Throughput

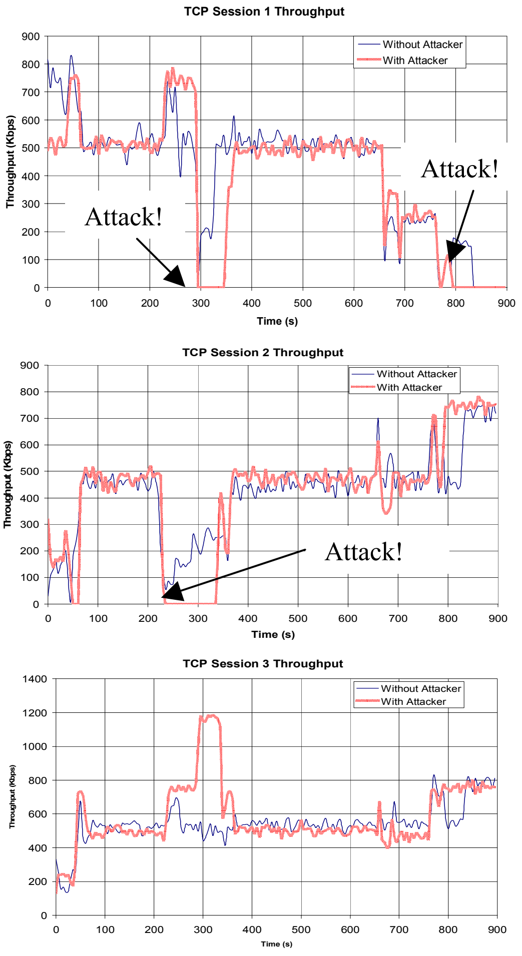
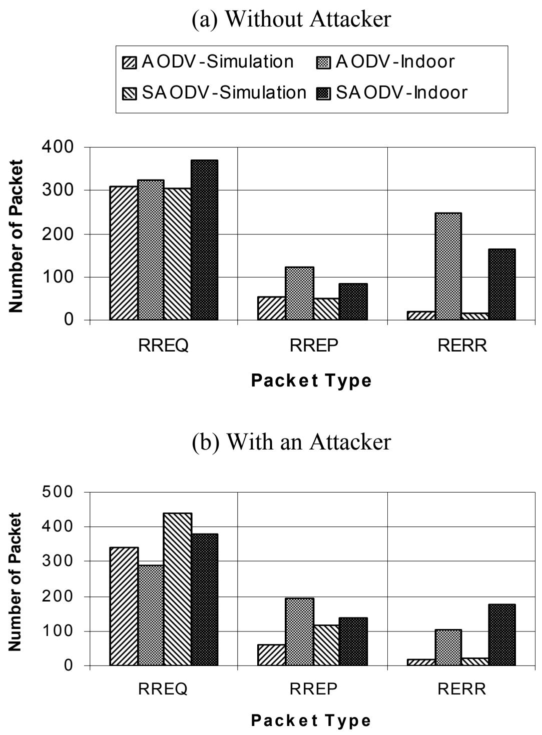
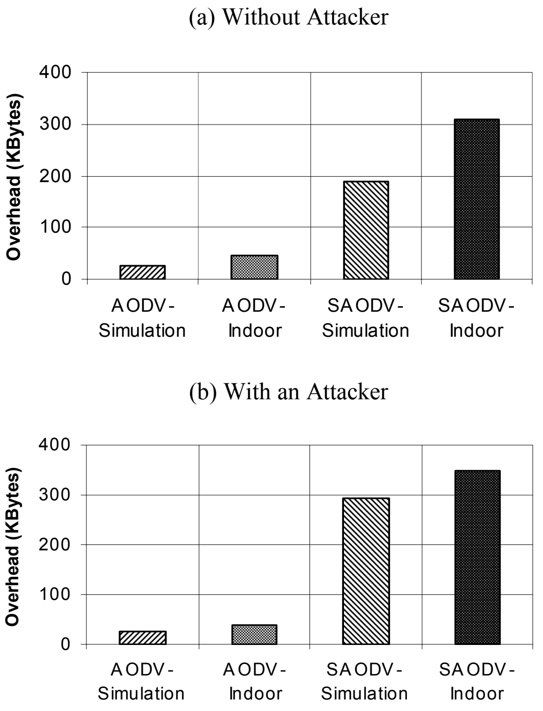


Figure 8. TCP Throughput under AODV (Indoor Emulation)


(a) Without Attacker

(b) With an Attacker

Figure 9. Routing Control Overhead (in Packets) for TCP


(a) Without Attacker

(b) With an Attacker

Figure 10. Routing Control Overhead (in bytes) for TCP

measured ratio of the number of data packets delivered to the destinations to the number of packets generated by all traffic sources. We also collected the statistics of the amount of *control overhead* (i.e., RREQ, RREQ, RERR) generated during each test run. Each time a control packet is forwarded, it is counted as one transmission. For TCP traffic, the *average throughput* is used.

4.1 Indoor Emulation and Simulation Results

4.1.1 UDP Traffic

Figure 4 shows the packet delivery fraction for the three sessions. When there is no attacker in the network, all three sessions show a high packet delivery ratio (i.e., above 90%) under both AODV and SAODV routing protocols. However, when there is an attacker, SAODV gives a higher packet delivery ratio than the original AODV under both simulation and indoor emulation tests. We notice the difference of the packet delivery ratio under AODV-indoor emulation and AODV-simulation. This is due to the fact that the indoor emulation neglects the real propagation model and assumes an ON/OFF wireless link status.

Figure 5 shows the amount of routing packets collected during the test run. Both simulation and indoor emulation results agree with each other. Since RREQ packets are re-broadcasted by many nodes while both RREP and RERR are sent by unicast, there is a higher ratio of RREQ packets than RREP and RREQ packets. The number of RREP packets increases in the presence of the attacker node because can attacker can send forge RREP to any received RREQ packet.

Figure 6 shows the routing control overhead (in bytes) collected during the test run. SAODV has a higher routing control overhead due to more additional fields in the RREQ and RREP packets. However, SAODV still gives a higher packet delivery ratio than AODV in the presence of an attacker.

Note that in SAODV, if each mobile node has the public key of all other nodes, then there is no need to carry the certificate information in the RREQ and RREP packets. The size of each control message can be reduced significantly (by 339 bytes).

4.1.2 TCP Traffic

Figure 7 shows the average TCP throughput for the three sessions. When there is no attacker in the network, session 3 has a higher throughput than sessions 1 and 2 under both simulation and indoor emulation. From the trace file, we noticed that both sessions 1 and 2 have a 2-hop path and share an intermediate node more often than session 3. There is a difference between TCP and UDP traffic in case of control packet tampering and data dropping attacks. For UDP traffic, any packet dropped by an attacker may never be recovered. However, the packet dropping attack may not be so effective to TCP flows especially in a mobile environment when the attacker may not be able to maintain itself in a location to be an intermediate node for a long period of time. Due to the size of the field we used in our testing (250 m by 100 m), the mobility trace generated in ns-2 for our tests puts the source and destination within transmission range for rather long proportion of time, which further weakens the attacker's ability to disrupt TCP performance.

Figure 8 shows the throughput performance via indoor emulation for the TCP sessions when using AODV with (or without) attackers. We can identify the periods of time when the attacker successfully blocked a flow's traffic. However, during those periods, other TCP flows gain higher throughput due to less traffic load in the network. As a result, the short service outage caused by one attacker may not necessarily lead to significant overall throughput decrease for a long TCP session. The effectiveness of the attack depends on the node movement patterns as well as the routing protocol's security features.

Figure 9 shows the amount of routing packets collected during the test run. Again, there is a higher ratio of RREQ packets than RREP and RREQ packets. There is also a higher percentage of RREP in case of routing attacks. Figure 10 shows the routing control overhead (in bytes) collected during the test run. Again, SAODV has a higher routing control overhead due to the additional fields in the RREQ and RREP packets. These results are similar to those shown in Figure 6 for UDP traffic.

Table 1. Gray Zone Measurement

Data Rate	d_1	d_2
2 Mb/s	110 m	9 m
11 Mb/s	100 m	18 m

4.2 Outdoor Results

4.2.1 Communication Gray Zone

Table 1 shows the average results we obtained via measurements on two different days. Results show that the width of the communication gray zone (i.e., d_2 in Figure 3) is about 9 m and 18 m under the transmission rate of 2 Mb/s and 11 Mb/s, respectively. When a node is within the gray zone, the link may not be declared as broken (i.e., HELLO packets may still be received occasionally, RERR may not be sent) even though multiple packet loss may occur.

When the transmission power is set to minimum, the transmission range with an average of more than 90% packet delivery ratio (i.e., d_1 in Figure 3) is about 110 m and 100 m under the transmission rate of 2 Mb/s and 11 Mb/s, respectively. Although this set of results may only be valid to the propagation environment of our outdoor test site, we are surprised by the measurement results that a reduction of the transmission bit rate (i.e., from 11 Mb/s to 2 Mb/s) only increases the transmission distance (i.e., d_1 in Figure 3) by 10%.

We attempted to further reduce the width of the gray zone by using the SNR threshold approach [28]. However, during our preparation, we noticed that the SNR value computed by the driver we used was actually the average value between all the neighboring links. That is, when multiple HELLO packets are received from multiple neighboring links, the device driver can only compute the average value rather than the instantaneous value for each received HELLO packet. Thus, the results presented in Sections 4.2.2 and 4.2.3 below may be subject to the communication gray zone problem.

4.2.2 UDP Traffic

Figure 11 shows the packet delivery ratio for each session. When there is no attacker in the network, all sessions (except session 2 for AODV) show a high packet delivery fraction under both AODV and SAODV routing protocols. Session 2 under AODV shows a lower packet delivery fraction. It may be due to the randomness in the users' movements and the communication gray zone problem. In each test case, the users' movement along the field may not be identical as in the previous test. When there is an attacker, SAODV gives a higher packet delivery fraction than the original AODV for all three sessions.

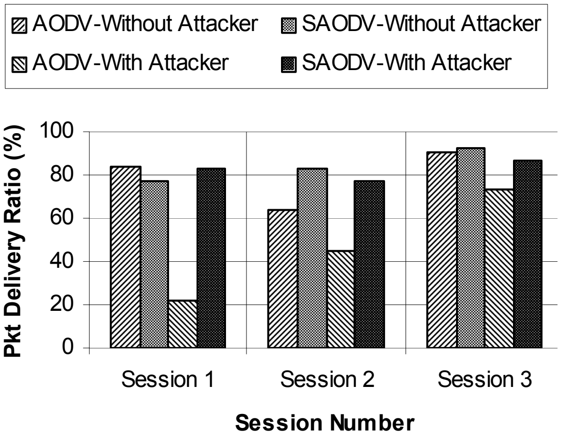
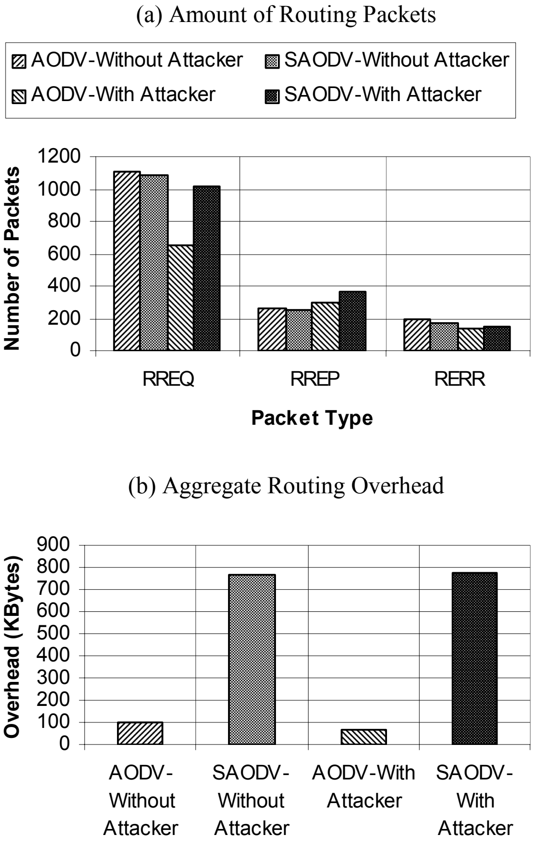


Figure 11. UDP Packet Delivery Ratio



(a) Amount of Routing Packets

(b) Aggregate Routing Overhead

Figure 12. Routing Control Overhead for UDP

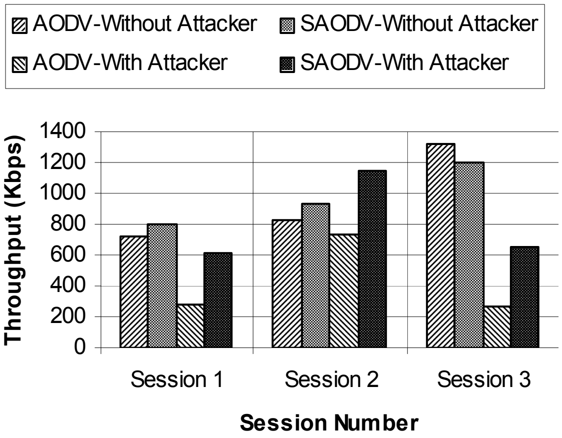
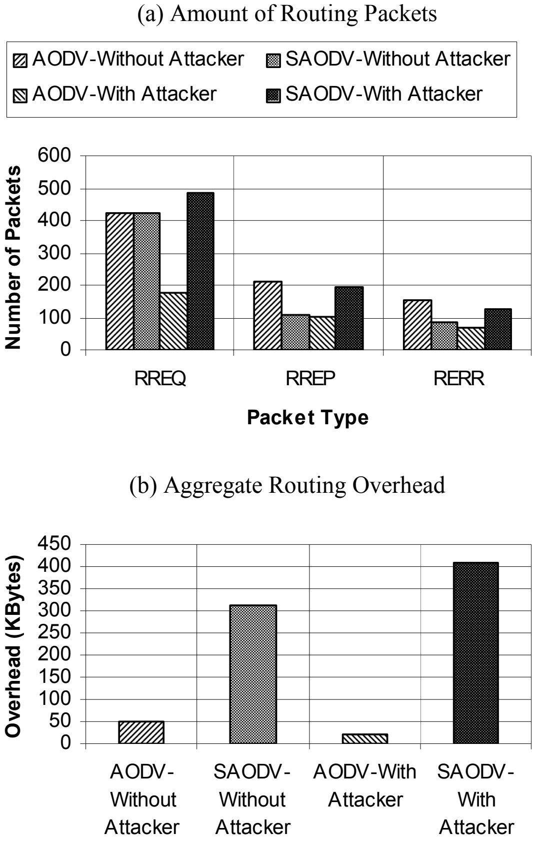


Figure 13. Average TCP Throughput



(a) Amount of Routing Packets

(b) Aggregate Routing Overhead

Figure 14. Routing Control Overhead for TCP

Figure 12 shows the amount of routing packets collected during the test run. Results show that SAODV does not introduce a significant increase in the number of transmitted control packets. However, due to the larger size in control packets for SAODV, the corresponding aggregate overhead (in bytes) for SAODV is higher than AODV. In spite of that, results in Figure 11 show that SAODV is effective in preventing control message tampering and data dropping attacks under UDP traffic.

4.2.3 TCP Traffic

Figure 13 shows the average TCP throughput for each session. Without an attacker, all three sessions show a high TCP throughput. The difference in throughput among individual sessions is due to the number of hops of the path during the test run. From the trace log file, we noticed that both sessions 1 and 2 have a higher fraction of time having a 2-hop path than session 3. When an attacker is present, the TCP throughput for all three sessions is decreased by more than 50%.

Figure 14 shows the amount of routing packets collected during the test run. Although SAODV has a higher control overhead due to the additional field to carry the certificate information, our results show that the extra control overhead does not decrease the TCP throughput significantly. These results show that SAODV is effective in preventing control message tampering and data dropping attacks under TCP traffic.

4.3 Discussions

Although we are able to compare the performance between AODV and SAODV routing protocols under different environments, our experiments have a number of limitations. The initial results presented in this paper only considered a single attacker. In future, we plan to conduct outdoor experiments and study the effect of an increased number of attackers.

Our current testbed with 10 laptops are considered to be small-scale. Other researchers have used between 20 and 40 laptops in their testbeds. A small testbed limits the path length and the number of available alternate paths between a source and destination pair. We plan to increase the number of laptops in our testbed incrementally over time.

Outdoor testing results presented in this paper are based on the data collected on a single test run. In future, we plan to repeat the experiments for several times and obtain the average values. We also plan to explore the impacts of enabling RTS/CTS and the auto-rate selection function.

Several problems were encountered when using the ad hoc mode. The AODV-UU version 0.8.1 provides an experimental function of using the received control packet signal strength to combat the gray zone. But this requires modification to the wireless card driver such that the received packet's signal strength value can be provided to routing modules. However, this functionality is not implemented in any driver by default because no layer above the PHY layer traditionally requires signal strength for individual packets.

For the new wireless cards used in our IBM T42 laptops, there is no native Linux driver support. The Madwifi driver which supports the Atheros chipsets is known for not working well in ad-hoc mode. The relatively stable driver we finally found to work in ad-hoc mode is the commercially available Driverloader working with the original Windows XP binary driver code. The lack of source codes for the driver makes it difficult to implement the signal strength support.

In addition, the commercial Driverloader driver is not always stable when working under ad-hoc mode. During the outdoor tests, when a user with the laptop walked away from other nodes and became disconnected to the ad-hoc network for a period of time, sometimes this laptop never managed to reconnect to the ad-hoc network even if it came within the range again. We managed to solve this problem by periodically refreshing the driver's ad-hoc mode setting during the tests.

Further work is required to improve the settings of the indoor experiments. For the indoor testing results presented in this paper, the nodes' movement traces and the link breakage information are obtained via simulation. The accuracy of the indoor testing results can be improved if the information is obtained via the outdoor testing experiments. In the outdoor testing experiments, the nodes' movement traces can be collected if there is a GPS (Global Positioning System) device attached to each laptop. The link breakage information can be collected in the trace file.

5. CONCLUSIONS

In this paper, we presented the quantitative performance comparisons between AODV and SAODV routing protocols under different environments (simulation, indoor emulation, and outdoor). We successfully implemented SAODV functionalities by modifying AODV-UU module. For outdoor testing, we built an experimental testbed with ten laptops under a 250 m by 100 m rugby field. Results under UDP and TCP traffic showed that SAODV is effective in preventing control message tampering and data dropping attacks. In addition, we also measured the communication gray zone area under different transmission bit rates. Results show that SAODV is effective in preventing control message tampering and data dropping attacks under UDP traffic.

For future work, we plan to compare the performance between other secure routing protocols (e.g., [8]). The number of nodes in our outdoor testbed will be increased (incrementally) over time. We also plan to study the performance improvement of cross-layer optimization between different layers (e.g., physical, MAC, network) in mobile ad-hoc networks under our outdoor testbed.

6. ACKNOWLEDGMENTS

The authors would like to thank Elizabeth Belding-Royer for her comments of our work during her visit at UBC in May 2005. The authors would also like to thank the following students who participated in the outdoor testing experiments in May and June 2005: Farshid Agharebparast, Syed Hussain Ali, Shahab Moradi, Enrique Stevens Navarro, Kevin Pang, Vijayanth Vivekanandan, Xiaolei Wang, Jeong-Jae Won, Walter Yin, and Limei Zhang. This work was supported by the Canada Foundation on Innovation (CFI) and the Canadian Natural Sciences and Engineering Research Council (NSERC) under grant RGPIN 261604-03.

7. REFERENCES

[1] C.E. Perkins, E. Belding-Royer, and S.R. Das, "Ad hoc On-demand Distance Vector (AODV) routing," *IETF RFC 3561*, July 2003.

[2] D. B. Johnson, D. A. Maltz, and Y.-C. Hu, "The Dynamic Source Routing protocol for Mobile Ad-hoc Networks (DSR)," *IETF Internet Draft* (*work in progress*), July 2004.

[3] X. Hong, K. Xu, and M. Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks," *IEEE Network*, vol. 16, issue 4, pp. 28-39, July/Aug. 2002.

[4] M. Mauve, J. Widmer, and H. Hartenstein, “A Survey on Position-based Routing in Mobile Ad Hoc Networks,” *IEEE Network*, vol. 15, issue 6, pp. 30-39, Nov./Dec. 2001.

[5] M. Zapata and N. Asokan, “Securing Ad-hoc Routing Protocols,” in *Proc. of ACM Workshop on Wireless Security (WiSe)*, Atlanta, GA, Sept. 2002.

[6] Y. Hu, A. Perrig, and D. B. Johnson, “Ariadne: A Secure On-demand Routing Protocol for Ad-hoc Networks,” in *Proc. of ACM MobiCom*, Atlanta, GA, Sept. 2002.

[7] S. Marti, T. J. Giuli, K. Lai, and M. Baker, “Mitigating Routing Misbehavior in Mobile Ad-hoc Networks,” in *Proc. of ACM MobiCom*, Boston, MA, Aug. 2000.

[8] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, “A Secure Routing Protocol for Ad-hoc Networks,” in *Proc. of International Conference on Network Protocols (ICNP)*, Paris, France, Nov. 2002.

[9] H. Yang, X. Meng, and S. Lu, “Self-organized Network-layer Security in Mobile Ad-hoc Networks,” in *Proc. of ACM Workshop on Wireless Security (WiSe)*, Atlanta, GA, Sept. 2002.

[10] P. Papadimitratos and Z. J. Haas, “Secure Routing for Mobile Ad-hoc Networks,” in *Proc. of SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, San Antonio, TX, Jan. 2002.

[11] Y. Hu, A. Perrig, and D. B. Johnson, “SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad-hoc Networks,” in *Proc. of IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, June 2002.

[12] P. Ning and K. Sun, “How to Misuse AODV: A Case Study of Insider Attacks Against Mobile Ad hoc Routing Protocols,” in *Proc. IEEE Information Assurance Workshop*, West Point, NY, June 2003.

[13] Y. Hu, A. Perrig, and D. B. Johnson, “Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols,” in *Proc. of ACM Workshop on Wireless Security (WiSe’03)*, San Diego, CA, Sept. 2003.

[14] S. Capkun and J.P. Hubaux, “BISS: Building Secure Routing out of an Incomplete Set of Security Associations,” in *Proc. of ACM Workshop on Wireless Security (WiSe’03)*, San Diego, CA, Sept. 2003.

[15] P. Papadimitratos and Z. J. Haas, “Secure Data Transmission in Mobile Ad Hoc Networks,” in *Proc. of ACM Workshop on Wireless Security (WiSe’03)*, San Diego, CA, Sept. 2003.

[16] J. Kong, X. Hong, Y. Yi, J.S. Park, J. Liu, and M. Gerla, “A Secure Ad-hoc Routing Approach using Localized Self-Healing Communities,” in *Proc of. ACM MobiHoc’05*, Urbana-Champaign, May 2005.

[17] D. Maltz, J. Broch, and D.B. Johnson, “Lessons from a Full-Scale Multihop Wireless Ad Hoc Network Testbed,” *IEEE Personal Communications*, pp. 8-15, Feb. 2001.

[18] C.K. Toh, M. Delwar, and D. Allen, “Evaluating the Communication Performance of an Ad Hoc Wireless Network,” *IEEE Trans. on Wireless Communications*, vol. 1, no. 3, pp. 402-414, July 2002.

[19] D. Aguayo, J. Bicket, S, Biswas, G. Judd, and R. Morris, “Link-level Measurements from an 802.11b Mesh Network,” in Proc. *ACM SIGCOMM’04*, Portland, Oregon, Sept. 2004.

[20] H. Lundgren, D. Lundberg, J. Nielsen, E. Nordstrom, and C. F. Tschudin, “A Large-scale Testbed for Reproducible Ad hoc Protocol Evaluations,” in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, March 2002.

[21] The Ad hoc Protocol Evaluation (APE) testbed, http://apetestbed.sourceforge.net/

[22] L. Klein-Berndt. Kernel AODV from National Institute of Standards and Technology (NIST). http://w3.antd.nist.gov/wctg/aodv kernel/

[23] V. Kawadia, Y. Zhang, and B. Gupta, “System Services for Implementing Ad-Hoc Routing: Architecture, Implementation and Experiences,” in *Proc. of the 1st International Conference on Mobile Systems, Applications, and Services (MobiSys),* San Francisco, CA, June 2003.

[24] I. D. Chakeres, AODV-UCSB Implementation from U. of California Santa Barbara, http://moment.cs.ucsb.edu/AODV/aodv.html.

[25] Windows XP based AODV from University of Bremen, http://www.aodv.org/

[26] C. Madson and R. Glenn, “The Use of HMAC-MD5-96 within ESP and AH,” *IETF RFC 2403*, Nov. 1998.

[27] The Network Simulator, http://www.isi.edu/nsnam/ns

[28] H. Lundgren, E. Nordstrom, and C. Tschudin, “Coping with Communication Gray Zones in IEEE 802.11b based Ad hoc Networks,” in *Proc. ACM International Workshop on Wireless Mobile Multimedia (WoWMoM’02),* Atlanta, Georgia, Sept. 2002.

[29] K.W. Chin, J. Judge, A. Williams, and R. Kermode, “Implementation Experience with MANET Routing Protocols,” *ACM SIGCOMM Computer Communications Review*, vol. 32, no. 5, pp. 49-59, Nov. 2002.

[30] R. Gray, D. Kotz, C. Newport, N. Dubrovsky, A. Fiske, J. Liu, C. Masone, S. McGrath, and Y. Yuan, “Outdoor Experimental Comparison of Four Ad Hoc Routing Algorithms,” in *Proc. ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Venice, Italy, Oct. 2004.

[31] ARAN (A Secure Routing Protocol for Ad Hoc Networks)’s implementation: http://signl.cs.umass.edu/arand/

[32] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit, “Ad Hoc Networking with Directional Antennas: A Complete System Solution,” *IEEE J. on Selected Areas in Communications*, vol. 23, no. 3, pp. 496-506, March 2005.

[33] R. Draves, J. Padhye, and B. Zill, “Routing in Multi-Radio, Multi-hop Wireless Mesh Networks,” in *Proc. ACM MobiCom’04*, Philadelphia, Pennsylvania, Sept./Oct. 2004.

[34] A. Raniwala and T.C. Chiueh, “Architecture and Algorithms for an IEEE 802.11-based Multi-Channel Wireless Mesh Network,” in *Proc. of IEEE Infocom’05*, Miami, Florida, March 2005.