Secure Routing with AODV Protocol for Mobile Ad Hoc Networks

Tahira Farid[↑] and Anitha Prahladachar[↑]
[↑] Department of Computer Science
[↑]University of Windsor

Abstract- **Mobile Ad Hoc Network (MANET) is a multi-hop wireless network of mobile nodes, forming a temporary network without the aid of any established infrastructure or centralized administration. Due to the absence of any dedicated routers, every node requires to contribute towards the configuration and maintenance of the routing framework. Since there are no centrally administered secure routers, attackers can easily exploit the network. Moreover, open peer-to-peer architecture, shared wireless medium, dynamic topology also adds on to the challenges in the security design of Mobile Ad Hoc Networks. These constrain make traditional secured routing schemes meant for wired networks unsuitable for mobile ad hoc environment. Routing in MANET is a challenging task receiving great amount of attention from researchers. Ad-hoc On-demand Distance Vector (AODV) is one the widely used routing protocols that is currently undergoing extensive research and development. In this paper we present the AODV protocol and survey various security enhancements that have been proposed for AODV by different researchers.**

Keywords- **Mobile Ad Hoc Networks, Routing, Security, AODV protocol.**

I. INTRODUCTION

Ad hoc networking is progressively becoming an important topic in the development of wireless technology moving towards the 4G network architecture (a network-of-networks intended to provide a variety of adaptable services to mobile and nomadic users by using integrated homogeneous architecture). Ad Hoc Network is a collection of independent nodes, corresponding to each other without a given fixed infrastructure. Therefore, they offer great flexibility, higher throughput, lower operating cost and better coverage compared to cellular base wireless networks. A wireless ad hoc network is primarily divided into two areas; Mobile Ad hoc Networks (MANET) and Smart Sensor Technology. Mobile ad hoc networks consist of mobile nodes, which can communicate with each other and nodes can enter and leave the network anytime. This dynamic nature brings in frequent topological changes in the network, making routing between mobile nodes a very difficult and challenging task. These challenges, along with the significance of routing protocols, make routing area the most active research area in the MANET domain.

Due to the short transmission range of Mobile Ad Hoc Networks, routes between nodes may consist of one or more hops. Thus each node may either work as a router or depend on some other node for routing. Figure 1.1 shows a simple ad hoc network with three mobile hosts using wireless interfaces. Host A and C are out of range from each other's wireless transmitter. When exchanging packets, they may use the

routing services of host B to forward packets since B is within the transmission range of both of them.

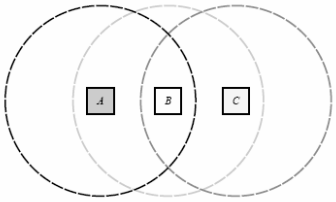

Figure 1.1: Mobile Ad hoc network with 3 mobile nodes

Mobile Ad Hoc Networks are useful in situations where geographical or terrestrial constrains demand totally distributed network system without fixed base station. Such situations can be in military battlefields or in any disaster and recovery situations. Due to such characteristics, these networks are highly susceptible to malicious attacks. They need harder security than conventional wired networks. Irrespective of the number of intrusion prevention schemes implemented in the Wireless Ad Hoc Network, there will be a vulnerable point in the network from which an intruder can break in. As the characteristics of Mobile Ad Hoc Networks are significantly different from wired network, well-established traditional security approaches to routing are inadequate for Mobile Ad Hoc Networks.

Routing protocols for Mobile Ad Hoc Networks can be broadly divided into two distinct categories, namely proactive (table-driven) routing protocols and reactive (on-demand) routing protocols. In Proactive Routing protocols, each node maintains up-to-date routing information to every other node in the network. Routing information is kept in a number of routing tables and updates to these tables are periodically transmitted throughout the network to maintain table consistency. Thus, in proactive routing, routes can be quickly established without any delay. However, it requires a significant amount of resources to keep routing information up-to-date.

Reactive or On-demand routing protocols are designed to overcome the increased overhead problem in proactive protocols. Unlike proactive protocols, reactive protocols create a route only when desired. If a node desires to send a message to a destination node for which it does not have a valid route to, it initiates a route discovery to locate the destination node. The process is completed when a source node finds a route to the destination. A route maintenance procedure is implemented to maintain a route until the destination is no longer available or not desired. Even though reactive protocols overcome increased overhead problem, but they exhibit end-to-end delay since routes are created on demand.

Both proactive and reactive routing protocols require persistent cooperative behavior, with intermediate nodes primarily contributing to the route development. Similarly each node, which practically acts like a mobile router, has absolute control over the data that passes through it. In essence, the membership of any ad-hoc networks indisputably calls for sustained depiction of benevolent behavior by all participating nodes [5]. This is often not possible in an open environment; this is the reason why these networks are frequently attacked by malicious nodes, from both inside and outside.

There are two kinds of possible attacks that can be initiated against Mobile Ad Hoc Networks: Passive and Active. In passive attacks the attacker does not disturb the routing protocol. It only eavesdrops upon the routing traffic and endeavors to extract valuable information like node hierarchy and network topology from it. In active attacks, malicious nodes can disturb the correct functioning of a routing protocol by modifying routing information, by fabricating false routing information, and by impersonating other nodes [2].

Generally cryptographic mechanisms are employed to protect routing protocols by enforcing mutual trust relationships among the wireless nodes [5]. Security in Mobile Ad Hoc Wireless Networks is mainly a dual problem. One is the security of the routing protocols that enable the nodes to communicate with each other and the second is the protection of the data that traverses the network on routes established by the routing protocols.

In this paper, we first discuss the traditional AODV routing protocol and the security flaws associated with it. Then we survey some of the secured approaches that have been proposed by different authors in order to secure AODV in a mobile ad hoc environment. We also investigate the experimental comparisons performed on the secured versions of AODV with the traditional AODV.

The rest of the paper is organized as follows. Section 2 describes the traditional AODV routing protocol and Section 3 and 4 discusses the security enhancements proposed for AODV. Experimental comparisons between AODV and secured AODV (SAODV) have been conversed in Section 5. Section 6 examines and evaluates the approached discussed in section 3, 4 and 5. Finally, Section 6 draws the conclusion.

II. AD HOC ON-DEMAND DISTANCE VECTOR ROUTING

Ad Hoc On-Demand Distance Vector (AODV) is a reactive or on-demand routing protocol. Reactive protocols are designed for Mobile Ad Hoc Networks to overcome the increased overhead problem in proactive protocols [4]. Unlike proactive protocols, reactive protocols create a route only when desired. If a node desires to send a message to a destination node for which it does not have a valid route to, it initiates a route discovery to locate the destination node. The process is completed when a source node finds a route to the destination. A route maintenance procedure is implemented to maintain a route until the destination is no longer available or not desired. Even though reactive protocols overcome increased overhead problem, but they exhibit end-to-end delay since routes are created on demand.

Route Discovery: On-Demand protocols employ a route discovery procedure, by which a source node discovers a route to a destination, for which it does not already have a route in its cache. The process broadcasts a ROUTE REQUEST packet, which is flooded across the network. In addition to the source node address and target node address, the request packet contains a route record, which records the sequence of hops taken by the request packet as it propagates through the network. RREQ

packets use sequence numbers to prevent duplication. The request is answered by a ROUTE REPLY packet either from the destination node or an intermediate node that has a cached route to the destination.

Route Maintenance: On-Demand protocols also employ a route maintenance procedure, where nodes monitor the operation of the route and inform the sender of any routing error. If a route breaks due to a link failure, the detecting host sends a ROUTE ERROR packet to the source, which upon receiving it, removes all routes in its cache that use the hop in error and initiates a new route discovery process.
AODV minimizes the number of broadcasts by creating routes on-demand as opposed to DSDV that maintains the list of all routes. The protocol is based on two phases, route discovery and route maintenance. A node does not perform route discovery or maintenance until it needs a route to another node or it offers its services as an intermediate node.

Local Hello messages are used to determine local connectivity, which can reduce response time to routing requests and can trigger updates when necessary. Sequence numbers are assigned to routes and routing table entries (used to supersede stale cached routing entries). Each node maintains two counters, node sequence number and broadcast ID. When a node wants to communicate with another node, but does not have a route to that node, it broadcasts a route request (RREQ) packet to its neighbors. The RREQ packet looks like Table 2.1, where source sequence number indicates the "freshness" of reverse route to the source; destination sequence number indicates the "freshness" of route to the destination; (source_addr, broadcast_id) uniquely identifies the RREQ.

Type	Flag	Resvd	hopcnt
Broadcast_id			
Dest_addr			
Dest_sequence_#			
Source_addr			
Source_Sequence_#			

Table 2.1: RREQ Packet of AODV

Every neighbor that receives the RREQ, either:
1. Returns a route reply packet (if route information about destination in its cache), or
2. Forwards the RREQ to its neighbors (if route information about destination not in its cache).

If a node cannot respond to the RREQ, the node increment the hop count, saves information to implement a reverse path set up (uses symmetric links because the route reply packet follows the reverse path of request packet). The information that are saved are: neighbor that sent the RREQ packet, destination IP address, source IP address, broadcast ID, source node's sequence number and expiration time for reverse path entry (to enable garbage collection).
For example, in Figure 2.1(a), node 1 needs to send a data packet to node 7, and let us assume that node 6 knows a current route to node 7 and no other route information exists in the network (related to node 7).

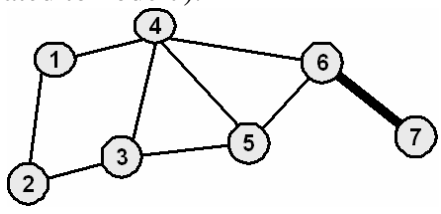

Figure 2.1(a): AODV Route Request

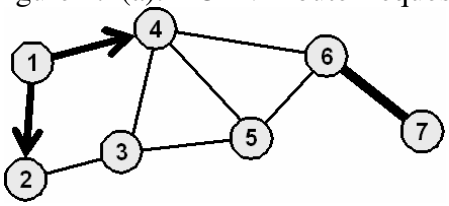

Figure 2.1(b): AODV Route Request

Node 1 sends a RREQ packet to its neighbors (Figure 2.1(b)):
Source_addr =1, dest_addr =7, broadcast_id = broadcast_id +1, source_sequence_# = source_sequence_# + 1, dest_sequence_# = last dest_sequence_# for node 7.

Nodes 2 and 4 verify that this is a new RREQ and that the source_sequence_# is not stale with respect to the reverse route to node 1. They forward the RREQ (Figure 2.1(c)), update source_sequence_# for node 1 and increment hop_cnt in the RREQ packet. RREQ reaches node 6 from node 4, which knows a route to 7. Node 6 must verify that the destination sequence number is less than or equal to the destination sequence number it has recorded for node 7. Nodes 3 and 5 will forward the RREQ packet to node 6, but it recognizes the packets as duplicates (Figure 2.1 (d)).

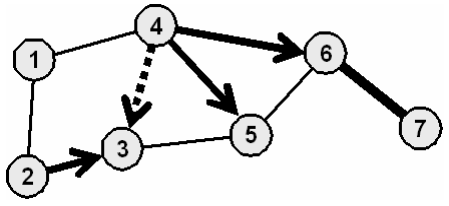

Figure 2.1(c): AODV Route Request

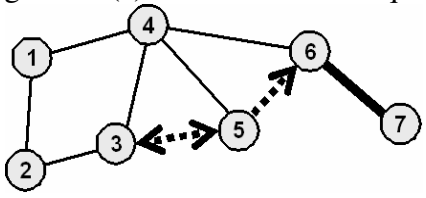

Figure 2.1(d): AODV Route Request

Now, if a node receives an RREQ packet and it has current route to the target destination, then it unicasts a route reply packet (RREP) to the neighbor that sent the RREQ packet. The RREP packet looks like Table 2.2.

Type	Flag	prsz	hopcnt
Dest_addr			
Dest_sequence_#			
Source_addr			
lifetime			

Table 2.2: RREP Packet of AODV

Intermediate nodes propagate the first RREP towards the source using cached reverse route entries. Other RREP packets are discarded unless, dest_sequence_# is higher than the previous, or dest_sequence_# is the same but hop_cnt is smaller (i.e. there is a better path). RREP eventually makes it to the source, which can use the neighbors sending the RREP as its next hop for sending to the destination. Also, cached reverse routes will timeout in nodes that do not see a RREP packet. For example, node 6 knows a route to node 7 and sends an RREP to node 4 (Figure 2.2 (a)):

Source_addr=1, dest_addr=7, dest_sequence_# = maximum (own sequence number, dest_sequence_# in RREQ), hop_cnt =1.

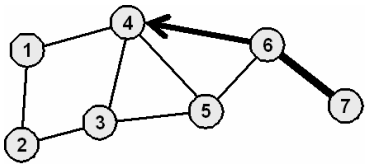

Figure 2.2(a): AODV Route Reply

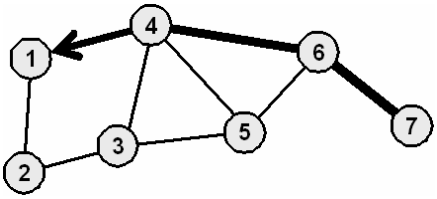

Figure 2.2(b): AODV Route Reply

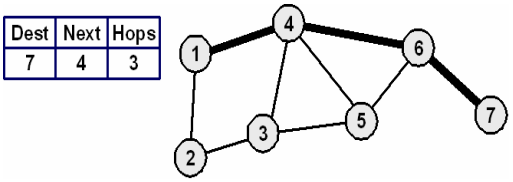

Figure 2.2(c): AODV Route Reply

Node 4 verifies that this is a new route reply (the case here), or one that has a lower hop count, and if so, propagates the RREP packet to node 1 (Figure 2.2 (b)). It also increments hop_cnt in the RREP packet. Node 1 now has a route to node 7 in three hops and can use it immediately to send data packets (Figure 2.2(c)). Therefore, the first data packet that prompted the path discovery has been delayed until the first RREP was returned.

Furthermore, Route changes can be detected by failure of periodic HELLO packets, failure or disconnect indication from the link level, or failure of transmission of a packet to the next hop (can detect by listening for the retransmission if it is not the final destination). The upstream (toward the source) node detecting a failure propagates a route error (RERR) packet to the source. The source (or another node on the path) can rebuild a path by sending a new RREQ packet.

This protocol is highly adaptive to dynamic networks but there is delay involved in route construction. Link breakage might begin another route discovery bringing in additional delays and consuming more bandwidth with the increase in the network size.

Since there is no protection for routing control packets and data packets in traditional AODV, many authors have proposed security ideas for AODV. Some of the ideas are discussed in the following sections.

III. SECURE ROUTING WITH THE AODV PROTOCOL

To protect Mobile Ad Hoc Networks from attacks a routing protocol must fulfill a set of requirements to ensure that the discovered path from source to destination functions properly in the presence of malicious nodes [2]. Some of such requirements that are addressed in [5] are as follows:

1) *Authorized nodes should perform route computation and discovery.*
2) *Minimal exposure to network topology.*
3) *Detection of spoofed routing messages.*
4) *Detection of fabricated routing messages.*
5) *Detection of altered routing messages.*
6) *Avoiding formation of routing loops.*
7) *Prevent redirection of routes from shortest paths.*

The major vulnerabilities present in the AODV protocol are:

1) *Deceptive incrementing of Sequence Numbers:* Destination Sequence numbers determine the freshness of a route. The destination sequence numbers maintained by different nodes are only update when a newer control packet is

received with a higher sequence number. However, a malicious node can increase this number in order to advertise fresher route to a particular destination.

2) *Deceptive decrementing of Hop Count:* AODV prefers route freshness over route length. A node would prefer a control packet with a larger destination sequence number and hop count over a control packet with a smaller destination sequence number and hop count. However, in case where the destination sequence numbers are same for two control packets, the route with the smaller hop count is chosen. A malicious node can easily exploit this mechanism by decrementing the Hop Count to generate fallacious smaller routes to destination.

In order to secure AODV, authors in [5] have divided the protocol into the following three categories:

1) *Key Exchange*
2) *Secure Routing*
3) *Data Protection*

1) *Key Exchange:* Most Key Exchange Protocols rely upon a central trust authority for initial authentication. A variant of the central trust authority is the Distributed Public-Key Model that makes use of threshold cryptography to distribute the private key of the Certification Authority (CA) over a number of servers. However, the requirements of a central trust authority in such a dynamic environment are considered impractical as well as unsafe. This is because an entity may not always be accessible and it also creates a single point of failure. Key Exchange using Key Distribution Server also poses similar problems.

Authors in [5] proposed that before entering the network, all nodes should obtain a one-time public and private key pair from the CA as well as the CA's public key. After that, nodes can negotiate session keys among each other, without any reliance on the CA, using any suitable key exchange protocol for Ad Hoc Networks without any dependence on the CA. These session keys are useful for securing the routing process and consequently the data flow. In order to avoid multiple peer-to-peer encryptions during broadcast of multicast operations, a group session key may be established between immediate nodes using a suitable Group Keying Protocol.

In [3] authors proposed the idea that during a group formation, the immediate neighboring nodes should engage in a shared RSA key generation procedure. They should generate a threshold sharing of an RSA key pair. This shared RSA key is used to provide the distributed group membership management and keying function [3].

A player *Pi* first obtains the group session key *Gk* used for group communications by combining a threshold *t* of partial RSA signatures/decryptions on a public value. The *public value* chosen may, for example, be the group name and time or session key number. Let *h(PublicValue)* represent the *pre-image* of the group key. The output of the threshold RSA signature protocol on this value is the group key. This can be computed anywhere in the system by any *t* players in the system [3].

This mechanism absolved the Ad Hoc Network of superfluous requirements

and provides necessary elements to secure both routing and data in presence of malicious nodes by providing authentication, non-repudiation, confidentiality and integrity.

2) *Secure Routing:* The main security problems linked to Ad Hoc Networks originate due to the route development by the intermediate nodes. It is therefore, imperative that only authorized nodes are allowed to update routing packets and malicious nodes are to be avoided at all costs. Peer-to-peer symmetric encryption of all routing information is has been proposed by authors in [5] to restrict modification of routing packets by intermediate nodes. All routing control packets between nodes are first encrypted and then transmitted. The route discovery and route maintenance procedures are described below:

During the Route Discovery Process, any node 'x' desiring to establish communication with node 'y' first establishes a group session key *Kx* with its immediate neighbors (Figure 3.1) and then creates the RREQ packet as in the AODV specification (Table 2.1).

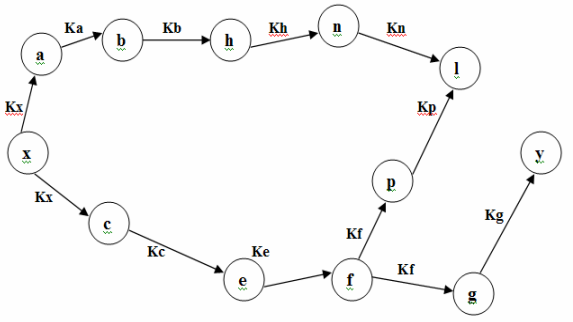

Figure 3.1: Point-to-point Establishment of Secure Routes

The RREQ packet is then encrypted using the group session key *Kx* and broadcasted. All intermediate recipient nodes that share the same group session key decrypt the RREQ packet, and if required, modify it according to the routing protocol specifications.

The intermediate nodes that do not possess group session keys with their immediate neighbors, initiate the group session key exchange protocol. After establishing the group session key, the intermediate nodes encrypt the RREQ packet using the new session key and rebroadcast the packet. This process follows until the packet reaches the final destination node 'y'.

After receiving the RREQ packet, to start the Route Reply Process, node 'y' creates a RREP packet as in the AODV specification (Table 2.2). The RREP packet is encrypted using the last group session key (*Kg* in this case) that was used to decrypt the received RREQ packet and is unicast back to the original sender. If any of the intermediate nodes has moved out of the wireless range a new group session key is established.

All recipient nodes that share the forward group session key decrypt the RREP packet and, if required, modify it according to the routing protocol specifications. The RREP packet is then again encrypted using the backward group session key and unicast to node 'x'. This process continues until the packet is received by node 'x'.

Each node in the network also maintains a table indexed by node ID as the primary key with associated group members and session keys (Figure 3.3) to avoid key synchronization problem. The table helps to establish secure routes with other nodes and a chain can be established using the available session keys. A secure key in Figure 3.3 is

highlighted between node ‘x’ and node ‘y’.

Source \ Destination	a	b	c	e	f	g	h	l	n	p	x	y
a		K_a									K_x	
b	K_a						K_h					
c				K_c							K_x	
e			K_c		K_e							
f				K_e		K_f				K_f		
g					K_f							K_g
h		K_h							K_h			
l									K_n	K_p		
n							K_h	K_n				
p					K_f			K_p				
x	K_x		K_x									
y						K_g						

Figure 3.3: Session Key Table [5]

In the process of Route Maintenance, all messages associated with route maintenance also need to be authenticated and protected from eavesdropping. A node which detects a broken link creates a RERR packet as in the AODV specification. The packet is then encrypted using a group session key in the direction of the recipient node using the session Key Table and is multicast back to the recipients.

Like Route Discovery, if any of the intermediate nodes moved out of the wireless range, a new group session key is established. All recipient nodes that share the group session key decrypt the RERR packet, and if required, modify it according to the routing protocol specifications. The RERR packet is then again encrypted using the group session key and is multicast back to the recipients. This process continues until the intended recipients receive the RERR packet.

3) *Data Protection:* Once protected routes have been established, in order to secure data transfer, any node ‘x’ desiring to establish an end-to-end secure data channel, first establishes a session key *Kxy* with the intended Node ‘y’ using the key exchange protocol as shown in Figure 3.4.

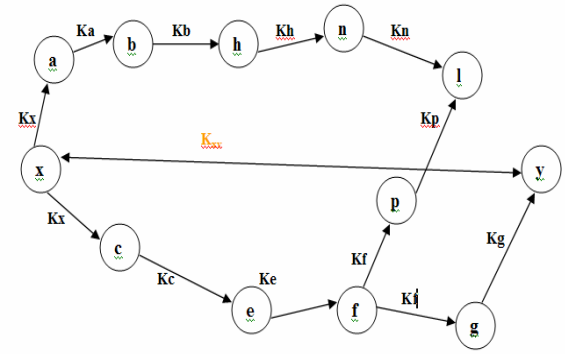

Figure 3.4: End-to-End Establishment of Secure Routes

Node ‘x’ then symmetrically encrypts the data packet using the session key *Kxy* and transmits it over the secure route. The intermediate nodes simply forward the packet in the intended direction without modifying anything. When the encrypted data packet reaches the destination it is decrypted using the session key *Kxy*. This process continues for all further data communication.

This approach provides authentication through the key exchange and all other services like confidentiality, integrity and non-repudiation rely on the accuracy of the authentication service. Following are the seven requirements that this approach satisfies:

1) Authorized nodes to perform route computation and discovery: Due to the authentication and key exchange protocol provided, the approach ensures that only authorized nodes are able to perform the route discovery. Malicious nodes will not be able to create fallacious routing packets as the routing control packets are all encrypted and authenticated by each intermediate node.

2) Minimal exposure of network topology: As all routing information is encrypted between nodes, a malicious node cannot gain any information regarding the network topology using passive eavesdropping.

3) Detection of spoofed routing messages: As the initial authentication relates a number of identities to each node's private key, the spoofing node need to create a similar private key in order to launch any attack.

4) Detection of fabricated routing messages: To fabricate a routing message the session key needs to be compromised, which is impossible as long as the key exchange protocol is assumed to be secure.

5) Detection of altered routing messages: Routing messages are relayed between the nodes in an unintelligible format. If the symmetric cipher also provides the integrity then the alteration of routing messages is virtually impossible.

6) Avoiding formation of routing loops: The proposed scheme ensures that routing loops cannot be formed through malicious action. It is possible otherwise if a malicious node is able to spoof, alter or fabricate legitimate routing packets.

7) Prevent redirection of routes from shortest paths: The scheme is designed in such a manner that routing packets are only accepted from authenticated immediate neighbors. This ensures that an adversary cannot inject such routing packets unless an authorized node first authenticates that particular node.

IV. SECURITY ENHANCEMENTS IN AODV PROTOCOL

Two types of security threats to the existing AODV protocol are described [6]

- *Internal attacks:* Internal attacks comprise of attacks by compromised nodes and selfish nodes. Compromised nodes are the nodes that are inside attackers who are behaving maliciously but can be authenticated by the network as a legitimate node and are being trusted by the other nodes. Selfish nodes are the nodes that tend to deny providing services for the benefit of other nodes in order to save their own resources.

- *External attacks:* External attacks comprise of attacks by malicious nodes. Malicious nodes are the attacker nodes which cannot authenticate themselves as legitimate nodes due to the lack of valid cryptographic information.

The model proposed in by authors of [1] to handle security attacks comprise of :

1) Intrusion Detection Model (IDM)
2) Intrusion Response Model (IRM)

Intrusion Detection Model (IDM):
Each node employs the detection model that utilizes the neighborhood information to detect misbehaviors of its neighbors (shown in the Figure 4.1)
When the misbehavior count for a particular node has reached its predefined threshold, the information is sent out to other nodes about the misbehaving node. The nodes receive it, check their local *malcount* for the broadcasted malicious node and add their result to the initiator's response.

The IDM is present on all the nodes. It constantly monitors the behavior of its neighbors and analyzes it to detect if the neighbor has been compromised. Four types of attacks addressed by authors of [1] are:

a) Distributed false route request
b) Denial of service
c) Destination is compromised
d) Impersonation

a) Distributed false route request:
A route request is generated whenever a node has to send data to the particular destination. A malicious node might generate frequent, unnecessary route requests. Moreover if a malicious node node generates a false route message from different radio range, it will be difficult to identify the malicious node. Route request messages are broadcast messages. When the node in the network receive a number of route requests that is greater than a threshold count by a specific source for a destination in a particular time interval *tinterval,* the node is declared as malicious and the information is propagated in the network.

b) Denial of service:
A malicious node launches the denial of service attack by transmitting false control packets and using the entire network resources. Thereby other nodes are deprived of the resources. Denial of service can be launched by transmitting false routing packets or data packets. It can be identified if a node is generating the control packets that are more than the threshold count in a particular time interval *tfrequency.*

c) Destination is compromised:
A destination might not be able to reply, if it is (i) not in the network; (ii) overloaded; (iii) it did not receive route request; or if it is (iv) malicious. This attack is identified when the source does not receive the reply from the destination in a particular time interval *twait.* Furthermore the neighbors generate *probe/ hello* packets to determine connectivity. If the node is in the network and does not respond to route requests destined for it, it is identified as malicious.

d) Impersonation:
It can be avoided if sender encrypts the packet with its private key and other nodes decrypts with the public key of the sender. If the receiver is not able to decrypt the packet, the sender might be not the real source and hence packet will be dropped.

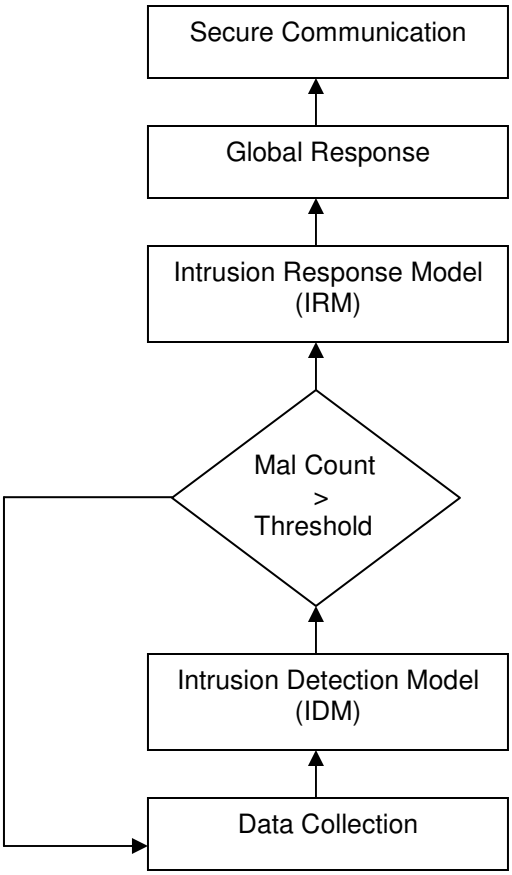

Figure 4.1: Handling of attacks [1]

Intrusion Response Model (IRM):

A node identifies that another has been compromised when its *malcount* increases beyond the threshold value for that allegedly compromised node. In such cases, it propagates this information to the entire network by transmitting *Mal* packet. If other nodes also suspect that the node that has been detected as compromised, it reports its suspicion to the network and transmits *ReMal* packet. If two or more nodes report about a particular node, *Purge* packet is transmitted to isolate the malicious node from the network. All nodes that have a route through the compromised node look for newer routes. All packets received from the compromised node are dropped.

V. EXPERIMENTAL COMPARISONS OF AODV AND SAODV ROUTING PROTOCOL

SAODV (Secure AODV)

The SAODV routing protocol proposed in [6] 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 $h^{hop\text{-}count}(seed)$, 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.

Three types of security threats to Mobile Ad-Hoc Networks have been addressed in [6]:

a) 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.

b) 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.

c) Message replay (or wormhole) attack: Attackers can retransmit 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.

We describe the hardware platform and the approach used to implement SAODV. We then describe the setup for simulation, indoor emulation, and outdoor experiments as conducted in University of British Columbia [6]

A. 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, default configuration for the radio interface is used. In all three experiments (simulation, indoor emulation, outdoor), the auto-rate selection and RTS/CTS are disabled.

B. Software Infrastructure

For AODV, we use the AODV-UU implementation 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. The hash chain functionalities for hop-count verification in the RREQ, RREP message handling modules was included. For the purpose of protecting routing messages with digital signatures, part of the code from the ARAN (Authenticated Routing for Ad-Hoc Networks) implementation was ported. 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, the intermediate nodes' capability of sending RREPs was disabled. Only the route destination node will send a signed RREP message. Attacker module was included 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.

C. 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.

D. Simulation Experiments

The ns-2 is used for the simulation experiments. The simulation time for each test is 1800 seconds. 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 was logged to emulate the mobility pattern later used for the indoor emulation tests.

E. Indoor Emulation Experiments

Figure 5.1: Indoor testing [6]

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, a mobility emulator MacSim which is similar to the MacKill program used in the APE project was implemented. Unlike the MacKill in APE which runs as a kernel module for packet killing, the well developed packet filtering program "iptables" in Linux for filtering packets based on source MAC address to emulate the link breakage was utilized.

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 5.1). 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, authors do mention that their current MacSim program can only simulate an ON/OFF binary state of the wireless link which may not be realistic in a wireless environment. More realistic results can be obtained if the mobility trace file is obtained via actual outdoor testing.

F. Comparison of test results for Indoor experiments:

For UDP traffic, the *packet delivery fraction* is defined as the 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.

UDP Traffic

Figure 5.2 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.

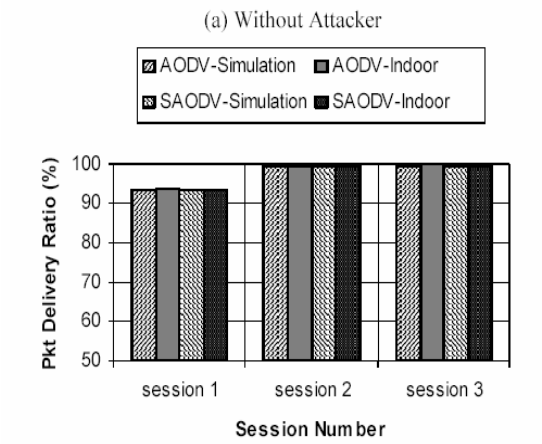

Figure 5.2(a) : Packet Delivery Ratio [6]

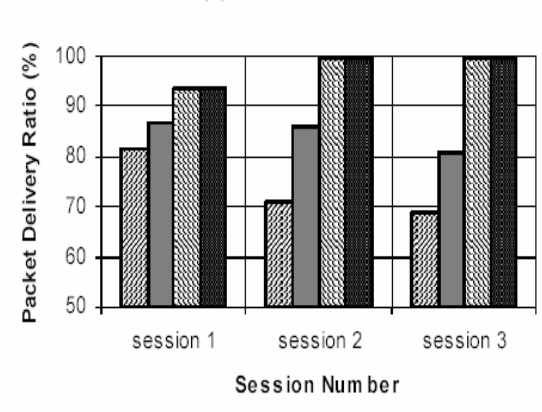

Figure 5.2(b) : Packet Delivery Ratio [6]

Figure 5.3(a) 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.

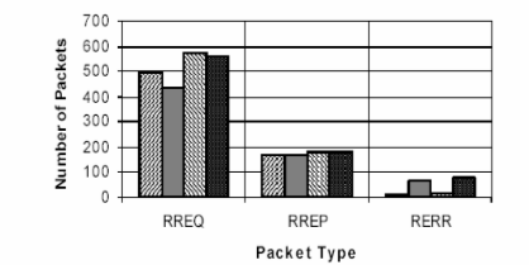

Figure 5.3(a): Number of Packets [6]

Figure 5.3(b) 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.

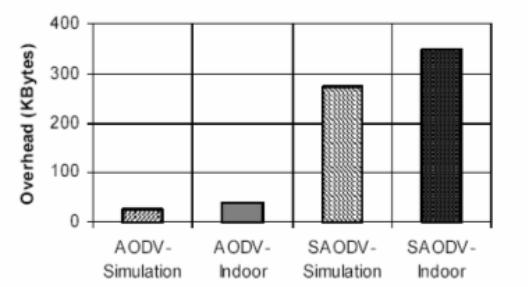

Figure 5.3(b): Overhead in bytes [6]

TCP Traffic

Figure 5.4 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, it is 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 the 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.

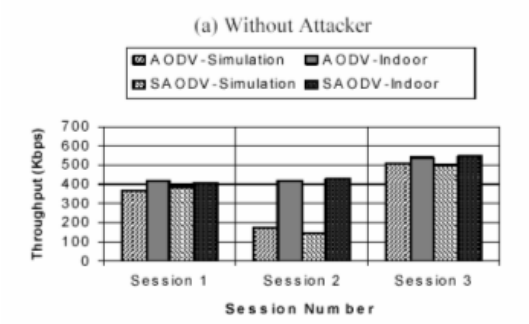

Figure 5.4(a): Throughput in Kbps [6]

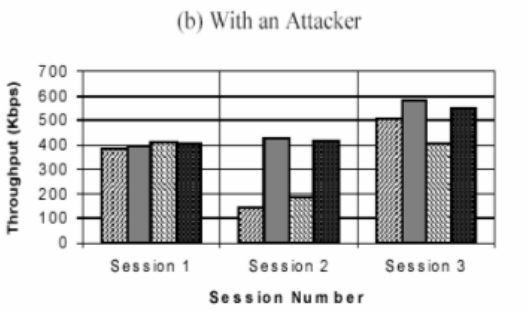

Figure 5.4(b): Throughput in Kbps [6]

Figure 5.5 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.

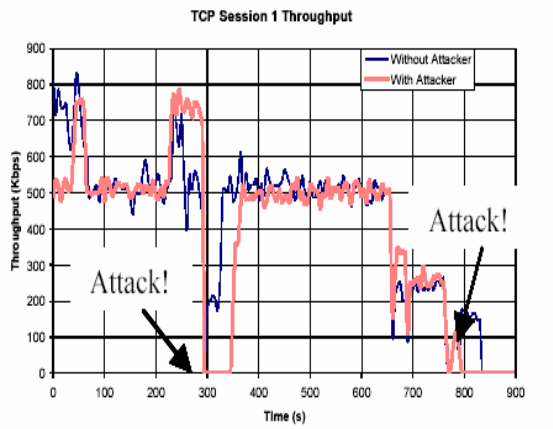

Figure 5.5(a): Session1 TCP [6]

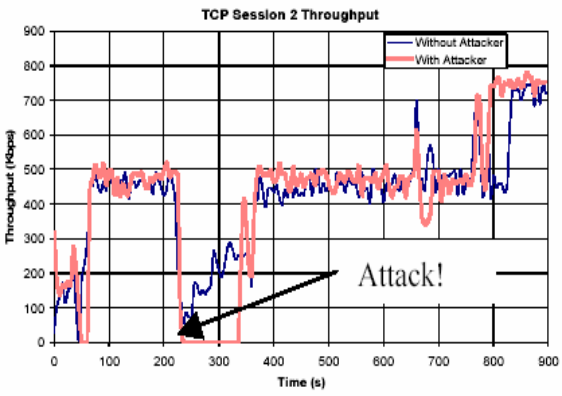

Figure 5.5(b): Session1 TCP [6]

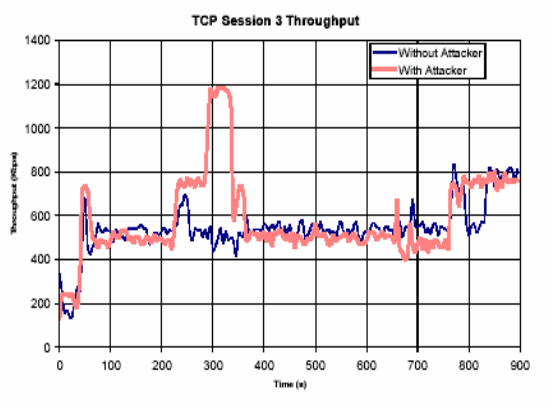

Figure 5.5(c): Session1 TCP [6]

Figure 5.6 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 5.7 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.

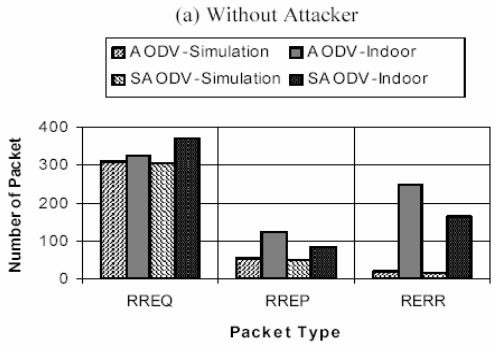

Figure 5.6(a): Packets collected [6]

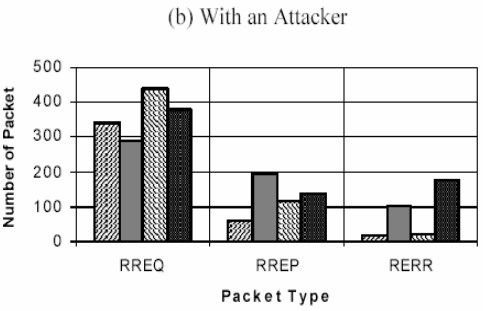

Figure 5.6(b): Packets collected [6]

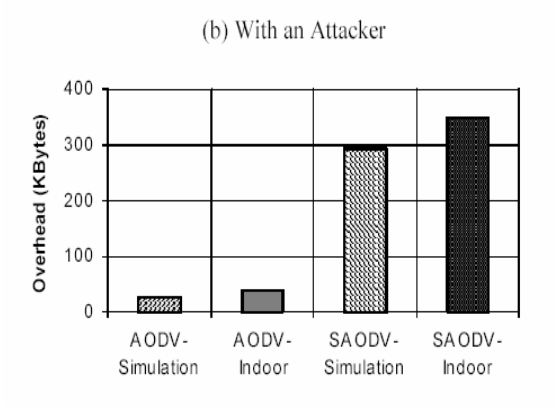

Figure 5.7: Overhead in KBytes [6]

G. Outdoor Experiments

The outdoor experiments were conducted in a rugby field which is near the university campus [6]. The area consists of ¼ sparsely clustered 3-story buildings and 3/4 open air field. Satellite images were used to confirm that the size of field is approximately 250 m by 100 m. Part of the outdoor field is shown in Figure 5.8. 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.

Figure 5.8: Outdoor field [6]

H. Comparison of test results for Outdoor experiments:

UDP Traffic

Figure 5.9 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 [6]. 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.

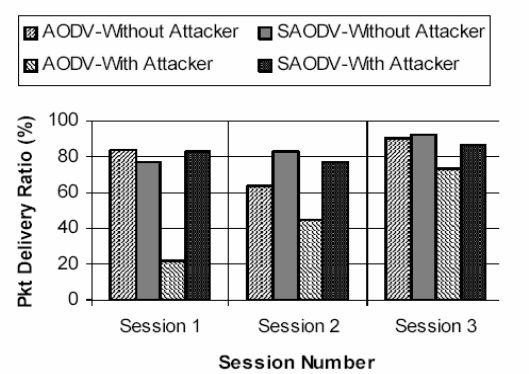

Figure 5.9: Packet Delivery ratio [6]

Figure 5.10 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 as shown in Figure 5.11. In spite of that, results in Figure 5.9 show that SAODV is effective in preventing control message tampering and data dropping attacks under UDP traffic.

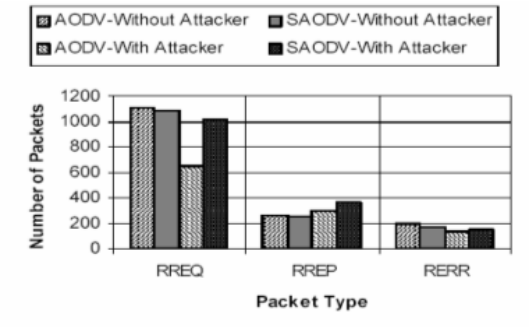

Figure 5.10: Packets collected [6]

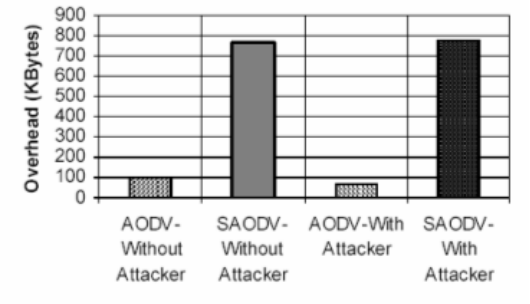

Figure 5.11: Packets collected [6]

TCP Traffic

Figure 5.12 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 5.13(a) shows the amount of routing packets collected during the test run. Although SAODV has a higher control overhead as shown in Figure 5.13(b) 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.

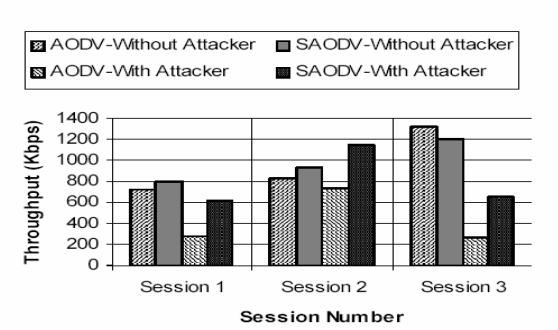

Figure 5.12: Throughput in Kbps [6]

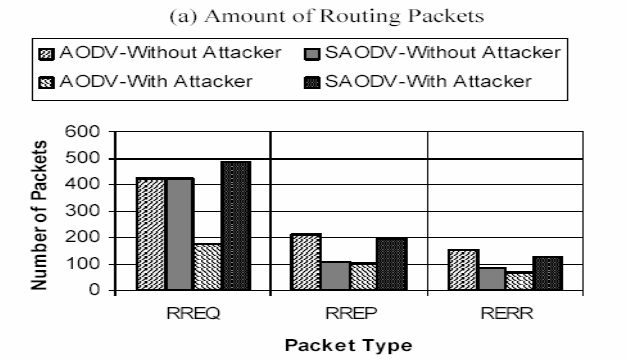

Figure 5.13(a): Packets collected [6]

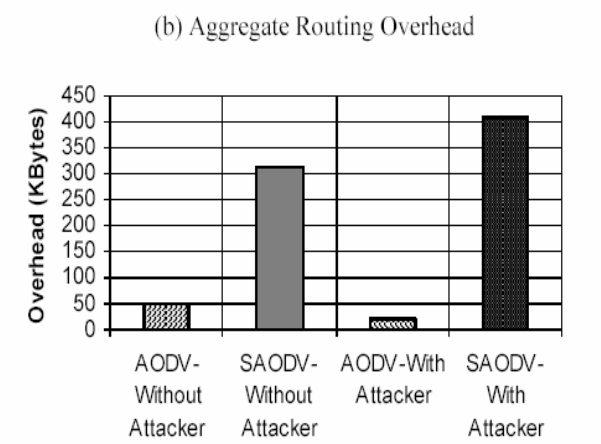

Figure 5.13(b): Packets collected [6]

VI. TESTING AND VALIDATION

The paper we discussed in Section III provided no testing procedure. We believe that since the paper was published very recently, authors are still working on the testing part of the protocol. However, after reviewing the paper we can conclude that the control packet and the data packet - protection provided by the approach seems very promising, but the robustness and feasibility of the protocol implementation still lies on the future testing.

The authors of paper we discussed in Section IV compared their secured version of AODV with the traditional AODV. The performance criteria were routing load, end-to-end delay, throughput and accuracy of prediction. The routing load of the network was increased by the malicious nodes as they generated false control messages. After implementing the proposed security model, it decreased the routing load than as in traditional AODV by identifying the malicious nodes and isolating them from the network. The end-to-end delay increased than AODV because of the time spent by the nodes to check for the malicious nodes and the throughput remained unaffected as compared to AODV. Moreover, the security model only predicted malicious nodes and other nodes were not accused of misbehaving.

Even though the protocol proposed to detect malicious nodes through the Intrusion Detection Model and Intrusion Response Model, no concrete approach has been provided to secure the data transmission.

The authors of paper discussed in Section V have conducted both Indoor and Outdoor testing procedure in great detail. The testing results have been compared to provide an objective overview of comparison between the performance of AODV and SAODV under similar conditions of 'without an attacker' and 'with an attacker'. The results provide a very clear set of information regarding the pros and cons of implementing SAODV in the place of AODV in today's Mobile Ad-Hoc Networks.

VI. CONCLUSION

We have surveyed research papers on fundamentally the AODV protocol for MANET and different approaches to secure AODV since the first version of traditional AODV had security vulnerabilities that could pose serious threat to data and control packets transmitted via MANETs which have quickly become most widely implemented and deployed in the world of Internet.

Such kind of literature surveys on security issues of existing protocols are of utmost importance as there is a persistent race between security threat causers and researchers intending to save networks from security attacks. This survey has provided us with a deep insight into the current status of attempts to secure AODV in the best manner with the most efficient protocol available. It has given direction in which one shall ideally work to do further research in the above mentioned domain.

REFERENCES

[1] Bhargava, S; Agarwal, D.P., *"Security Enhancements in AODV protocol for Wireless Ad Hoc Networks,"* In IEEE VTS conference on Vehicular Technology, Vol. 4, October 7-11, 2005.

[2] Dahill, B.N.; Royer, E.; Shields, C., *"A secure Routing Protocol for Ad Hoc Networks,"* in proceedings of International Conference on Network Protocols (ICNP), pp. 78-87, 2002.

[3] Lehane, B.; Dolye, L.; O'Mahony, D., "Ad Hoc Key Management Infrastructure," in proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05), Vol. 2, pp. 540-545, 2005.

[4] Perkins, C.E.; Royer, E.M., *"Ad-hoc On-Demand Distance Vector Routing,"* in proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications, WMCSA, 1999.

[5] Pirzada, A.A.; McDonald, C, *"Secure Routing with the AODV Protocol,"* in proceedings of the Asia-Pacific Conference on Communications, October 3-5, 2005.

[6] Yuxia Lin, A. Hamed Mohsenian Rad, Vincent W. S. Wong, Joo-Han Song, *"Experimental Comparisons between SAODV and AODV Routing Protocols,"* in proceedings of the 1st ACM workshop on Wireless Multimedia Networking and Performance modeling, WMuNeP, October 2005.