Impact of AODV routing protocol for establishing Security in Wireless Ad Hoc Networks in Indoor Environments

Seyed Amin Ahmadi Olounabadi[1], Prof. Avula.Damodaram[2], Prof. V Kamakshi Prasad[3]

ABSTRACT -- A mobile ad hoc network (MANET) is a continuously self-configuring, self-organizing, infrastructure-less network of mobile devices connected wirelessly. It is sometimes known as "on-the-fly" networks or "spontaneous networks, and also known as wireless ad hoc network or ad hoc wireless network. Instead of relying on a base station to coordinate the flow of messages to each node in the network, the individual network nodes forward packets to and from each other. Basically, an ad hoc network is a temporary network connection created for a specific purpose (such as transferring data from one computer to another). Multipath routing is the routing technique of using multiple alternative paths through a network, which can yield a variety of benefits such as fault tolerance, increased bandwidth, or improved security.

Ad hoc On-Demand Distance Vector (AODV) Routing is a routing protocol for mobile ad hoc networks (MANETs) and other wireless ad hoc networks. The AODV routing protocol is an On-Demand routing protocol in which all paths are only discovered when needed and only during the time they are used. When a node is discovered with a path to the destination, that path is reported back to the source node that requested the path. . AODV enables nodes to use multiple paths to discover destinations, and keep these paths up until the network topology changes in a consistent manner. They also deal strongly with routing loops because they are expensive on any network, especially on a wireless network with signalling capacity and Node processing power. AODV goals are: Minimum control overhead, Minimum processing overhead, Multiple-routing capability, Dynamic topology storage and Unavailability of loop.

Simulations were conducted using the NS2 network simulator. We are considering our communication path is changeable even path or node is node failed. So data is sending through different paths, it provide high security than single path.

Key words: Mobile Ad hoc Network (MANET), Multipath routing, AODV, SMT protocol, Indoor Environment, NS2 (Network Simulator V2).

[1] Ph.D. Scholar Student in Computer Science and Engineering, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad, Telangana, India
saminahmadi@hotmail.com

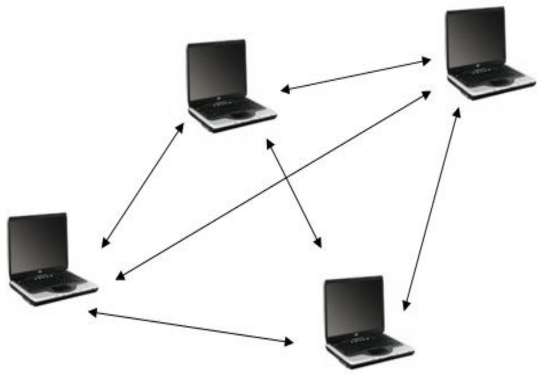
[2] Vice-Chancellor Sri Venkateswara University, Tirupati, Andhra Pradesh, India. Faculty of Computer Science & Engineering at (JNTUH), Hyderabad, Telangana, India
vcsvutpt@yahoo.com

[3] Director of Evaluation, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad, Telangana, India
kamakshiprasad@jntuh.ac.in

1 INTRODUCTION

Defining Ad hoc Network

A mobile ad hoc network (MANET) is a continuously self-configuring, self-organizing, infrastructure-less network of mobile devices connected wirelessly. It is sometimes known as "on-the-fly" networks or "spontaneous networks, and also known as wireless ad hoc network or ad hoc wireless network. Basically, an ad hoc network is a temporary network connection created for a specific purpose (such as transferring data from one computer to another). In Wireless ad hoc networks, network is dynamic and on it, nodes are free to move. Wireless networks lack the complexities of infrastructure setup and administration, enabling devices to create and join networks anywhere, anytime, and when there is a need for people to communicate using mobile devices. Since MANETS rely on wireless transmission, a secured way of message transmission is important to protect the privacy of the data. An insecure ad-hoc network at the edge of communication infrastructure may potentially cause the entire network to become vulnerable to security breaches. Therefore nodes must cooperate for their integrity of the operation of the network. However, nodes may refuse to cooperate due to not forwarding packets for others for selfish reasons and not want to expose their resources. In this proposed system, a fixed threshold is used to identify the faults.



The system can be compared with any of the multipath routing protocols. The additional delay due to probing might be reduced if the location of nodes after mobility especially destination node and adversaries can be predicted. This knowledge about nodes future location and behavior will be helpful in highly secured applications and also in pervasive computing where mobile ad hoc networks plays a major role.

SMT protocol provides security based on the security association between the end nodes. It is not able to overcome the compromised nodes attacks. The work presented in this paper has two phases. The first phase is to improve the security and reliability of data transmission in mobile ad hoc networks by providing secured routes.

1.1 Multipath routing

Multipath routing is the routing technique of using multiple alternative paths through a network, which can yield a variety of benefits such as fault tolerance (should a path fail, only the traffic assigned to that path is affected, the other paths continuing to serve their stream flows; there is also, ideally, an alternative path immediately available upon which to continue or restart the interrupted stream), increased bandwidth, or improved security. The multiple paths computed might be overlapped, edge-disjointed or node-disjointed with each other. Each stream is assigned a separate path, uniquely to the extent supported by the number of paths available. If there are more streams than available paths, some streams will share paths. This provides better utilization of available bandwidth by creating multiple active transmission queues.

1.2 Indoor Environments

Indoor environments, Set up by using a pair of wireless laptops to file share where there is human movement between the two nodes, Wi-Fi link throughput is measured in an obstructed office block, laboratory, library, and suburban residential home environments.

1.3 Mobile Ad-hoc Networks

An ad-hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any stand-alone infrastructure or centralized administration. Mobile Ad-hoc networks are self-organizing and self-re-configuring multi hop wireless networks where, the structure of the network changes dynamically. This is mainly due to the mobility of the nodes. Nodes in these networks utilize the same random access wireless channel, cooperating in a friendly manner to engaging themselves in multi hop forwarding. The nodes in the network not only act as hosts but also as routers that route data to/from other nodes in network.

In mobile ad-hoc networks a destination node might be out of range of a source node transmitting packets; a routing procedure is always needed to find a path so as to forward the packets appropriately between the source and the destination. Within a cell, a base station can reach all mobile nodes without routing via broadcast in common wireless networks. In the case of ad-hoc networks, each node must be able to forward data for other nodes. This creates additional problems along with the problems of dynamic topology which is unpredictable connectivity changes.

MANETS rely on wireless transmission, a secured way of message transmission is important to protect the privacy of the data. An insecure ad-hoc network at the edge of communication infrastructure may potentially cause the entire network to become vulnerable to security breaches. There is no central administration to take care of detection and prevention of anomalies in mobile ad hoc networks. Mobile devices identities or their intentions cannot be predetermined or verified. Therefore nodes have to cooperate for the integrity of the operation of the network. However, nodes may refuse to cooperate by not forwarding packets for others for selfish reasons and not want to exhaust their resources. Various other factors make the task of secure communication in ad hoc wireless networks difficult include the mobility of the nodes, a promiscuous mode of operation, limited processing power, and limited availability of resources such as battery power, bandwidth and memory.

Attacks on ad hoc are classified into non-disruptive passive attacks and disruptive active attacks. The active attacks are further classified into external attacks and internal attacks are carried out by nodes that do not belong to network and can be prevented by firewalls and encryption techniques. Internal attacks are from internal nodes which are actually authorized nodes and part of the network hence it is difficult to identify

Secure Message Transmission (SMT) protocol provides security based on the security association between the end nodes. It is not able to overcome the compromised nodes attacks. The work presented in this paper has two phases. The first phase is to improve the security and reliability of data transmission in mobile ad hoc networks by providing secured routes.

The current topological information will be gathered based on the network behavior such as transmission time, Probability of lost packets and correctly received – acknowledged packets and a threshold is set which is used in binary search probing. The nodes may exchange the current velocity vectors such as speed and direction to predict the location of the nodes. The spatial and temporal mining can be used to find the relative appropriateness of the location.

In ad hoc networks devices (also called nodes) act both as computers and routers. Most routing protocols lead nodes to exchange network topology information in order to establish communication routes. This information is sensitive and may become a target for malicious adversaries who intend to attack the network or the applications running on it. There are two sources of threats to routing protocols. The first comes from external attackers. By injecting erroneous routing information, replaying old routing information, or distorting routing information, an attacker could successfully partition a network or introduce a traffic overload by causing retransmission and inefficient routing. The second and more severe kind of threat comes from compromised nodes, which might (i) misuse routing information to other nodes or (ii) act on applicative data in order to induce service failures.

1.4 Problems with Routing In Mobile Ad-Hoc Networks

Asymmetric links: Most of the wired networks rely on the symmetric links which are always fixed. But this is not a case with ad-hoc networks as the nodes are mobile and constantly changing their position within network.

Routing Overhead: In wireless ad-hoc networks, nodes often change their location within network. So, some stale routes are generated in the routing table which leads to unnecessary routing overhead.

Interference: This is the major problem with mobile ad-hoc networks as links come and go depending on the transmission characteristics, one transmission might interfere with another one and node might overhear transmissions of other nodes and can corrupt the total transmission.

Dynamic Topology: another major problem with ad-hoc routing since the topology is not constant. The mobile node might move or medium characteristics might change. In ad-hoc networks, routing tables must somehow reflect these changes in topology and routing algorithms have to be adapted. For example in a fixed network routing table updating takes place for every 30sec. This updating frequency might be very low for ad-hoc networks.

1.5 Routing Protocols

1.5.1 Routing Tables

Each routing table entry contains the following information

- Destination
- Next hop
- Number of hops

- Destination sequence number
- Active neighbors for this route
- Expiration time for this route table entry. Expiration time, also called lifetime, is reset each time the route has been used. The new. Expiration time is the sum of the current time and a parameter called active route timeout. This parameter, also called route caching timeout, is the time after which the route is considered as invalid, and so the nodes not lying on the route determined by RREPs delete their reverse entries. If active route timeout is big enough route repairs will maintain routes. RFC 3561 defines it to 3 seconds.

1.5.2 Control messages

- Routing request

When a route is not available for the destination, a route request packet (RREQ) is flooded throughout the network. The RREQ contains the following fields:

Source addresses, request ID, source sequence number, destination address, destination sequence number, hop count.

The request ID is incremented each time the source node sends a new RREQ, so the pair (source address, request ID) identifies a RREQ uniquely. On receiving a RREQ message each node checks the source address and the request ID. If the node has already received a RREQ with the same pair of parameters the new RREQ packet will be discarded. Otherwise the RREQ will be either forwarded (broadcast) or replied (unicast) with a RREP Message.

- Routing reply

If a node is the destination, or has a valid route to the destination, it uncast a route. Reply message (RREP) back to the source. This message has the following format:

Source, destination address, destination sequence Number, hop count, life time.

The reason one can unicast RREP back is that every node forwarding a RREQ message: Caches a route back to the source node.

- Route error

All nodes monitor their own neighborhoods. When a node in an active route gets lost, a route error message (RERR) is generated to notify the other nodes on both sides of the link of the loss of this link.

- HELLO messages

Each node can get to know its neighborhoods by using local broadcasts, so-called HELLO messages. Nodes neighbors' are all the nodes that it can directly communicate with. Although AODV is a reactive protocol it uses these periodic HELLO messages to inform the neighbor's that the link is still alive. The HELLO messages will never be forwarded because they are broadcasted with TTL = 1. When a node receives a HELLO message it refreshes the corresponding lifetime of the neighbor information in the routing table. This local connectivity management should be distinguished from general topology management to optimize response time to local changes in the network.

1.6 Ad hoc On Demand Vector Routing (AODV)

The Ad-hoc On-Demand Distance Vector (AODV) routing protocol is designed for use in ad-hoc mobile networks. AODV is a reactive protocol: the routes are created only when they are needed. It uses traditional routing tables, one entry per destination, and sequence numbers to determine whether routing information is up-to-date and to prevent routing loops. An important feature of AODV is the maintenance of time-based states in each node: a routing entry not recently used is expired. In case of a route is broken the neighbors can be notified. Route discovery is based on query and reply cycles, and route information is stored in all intermediate nodes along the route in the form of route table entries. Link-state routing algorithms are more reliable, less bandwidth-intensive, but also more complex and compute- and memory-intensive. In on-demand routing protocols a fundamental requirement for connectivity is to discover routes to a node via flooding of request messages.

The AODV routing protocol is an On-Demand routing protocol in which all paths are only discovered when needed and only during the time they are used. The tracks are discovered during a flooding, during which Network nodes are questioned in the process of searching for a path to the destination. When a node is discovered with a path to the destination, that path is reported back to the source node that requested the path.

AODV goals are: Minimum control overhead, minimum processing overhead, Multiple-routing capability, Dynamic topology storage, Unavailability of loop.

Because resources on Ad-hoc mobile networks are scarce, AODV tries to minimize control overflow by limiting alternate path updates and also only using On-Demand messages. In order to minimize overhead, AODV message structures they are simple and require little computing. In an Ad-hoc network, resources and exams may be outside of the direct communication area, due to the limited scope of the sending of wireless equipment. Thus, AODV enables nodes to use multiple paths to discover destinations, and keep these paths up until the network topology changes in a consistent manner. They also deal strongly with routing loops because they are expensive on any network, especially on a wireless network with signalling capacity and Node processing power. AODV uses sequential numbers in each node to prevent routing loops. This protocol consists of two phases: 1- Route detection 2- Route maintenance.

AODV defines the following types of messages: 1- Route Request (RREQ) is broadcasted by a node requiring a route to another node, 2-Route Reply (RREP) is unicasted back to the source of RREQ, 3- Route Error (REER) is sent to notify other nodes of the loss of the link, 4- Route Reply Acknowledgment (RREP-ACK) HELLO messages are used for detecting and monitoring links to neighbors. In the path discovery process, when a source node needs a path to a destination node and the path is not valid in the routing table, the source node returns the origin of a path request packet (RREQ) to the destination node. When receiving each RREQ node, it creates or updates a reverse entry to the source node in the routing table, and if a valid path in the routing table to the destination node does not re-establish the RREQ again. When flooding the RREQ packet from the source node to the Destination Node, the Node creates or reverses the inbound route destination, and creates a single path response packet (RREP), which is an enlarged numbered set, in the opposite direction. When the RREP reaches the origin node and goes in the opposite direction, it creates or updates a forward path to the destination, and communications begin.

In the process of storing the path of each node, it alternates a Hello package for all local connections and generates RREP with TTL = 1, such as the Hello package. When a packet node does not receive a packet from a neighbour in a few seconds, it assumes that the connection to the neighbour has been broken. In addition, when a faulty connection node has a neighbour to the MAC layer notification, a failure in the path to the node Determines the destination that the next step of the path is the same ninety neighbour. When the node detecting the failure of the connection is close to the destination node (that is, the number of steps to the destination node is much smaller than the number of steps to the ninety of the source (requires a new path to the destination, which is called Local Repair, Is called

Local repair is also a path discovery process and is very similar to what was said in the previous sections. During the local repair, data packets are buffered. When the RREP reaches the local repair and succeeds, the Node starts sending the packets of the buffered data. When node detects that the connection failure is far from the destination node or when the local repair is not successful, Node publishes a path error packet (RERR) to the source node, including the destination address of the inaccessible. When each middle node receives the RERR, the routes that the destination node is inaccessible and the next step is NER, the RERR issuer is invalid, and the RERR is re-released. When the Node receives the RERR source, the path to the destination node also becomes invalid and re-locates the path.

2 SYSTEM ANALYSIS

2.1 Proposed system

In this proposed system, a fixed threshold is used to identify the faults. Instead of fixed threshold, varying threshold considering dynamic changing networks can be set. The system can be compared with any of the multipath routing protocols. The additional delay due to probing might be reduced if the location of nodes after mobility especially destination node and adversaries can be predicted. This knowledge about nodes future location and behavior will be helpful in indoor environments and also in pervasive computing where mobile ad hoc networks plays a major role. Also this work with little variations along with service oriented architecture can be adapted for providing privacy and trust in pervasive computing.

SMT protocol provides security based on the security association between the end nodes. It is not able to overcome the compromised nodes attacks. The work presented in this paper has two phases. The first phase is to improve the security and reliability of data transmission in mobile ad hoc networks by providing secured routes.

The faults are identified and those links will be avoided in the data transmission phase. The current topological information will be gathered based on the network behavior such as transmission time, Probability of lost packets and correctly received – acknowledged packets and a threshold is set which is used in binary search probing. The nodes may exchange the current velocity vectors such as speed and direction to predict the location of the nodes. The spatial and temporal mining can be used to find the relative appropriateness of the location.

3 METHODOLOGY

3.1 NS -2 Introductions

- It is Cheap and does not require costly equipment
- The real thing isn't yet available
- Complex scenarios can be easily tested.
- Controlled experimental conditions
    - Reusability helps aid debugging
- Results can be quickly obtained
    - More ideas can be tested in smaller timeframe.
- Disadvantages: Real systems are too complex to model

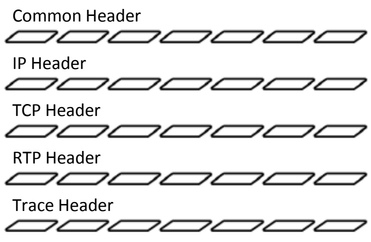
3.2 NS-2 Features

- NS-2 is an object oriented discrete event simulator
    - Single thread of control: no locking or race conditions
    - Simulator maintains list of events and executes in sequence order i.e.) one event after another
- Back end is C++ event scheduler
    - Protocols mostly
    - Fast to run, more control
- Front end is OTCL
    - Creating scenarios, extensions to C++ protocols
    - Fast to write and change

3.3 Ns-2 Programming Structure

- Create the event scheduler
- Turn on tracing
- Create network topology
- Create transport connections
- Generate traffic
- Insert errors
- *PACKETS*

  It is the collection of data, whether header is called or not all header files where present in the stack register



Packets Format

4 DESIGN

4.1 Modules

4.1.1 Mobile Ad-hoc Networks

An ad-hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any stand-alone infrastructure or centralized administration. Mobile Ad-hoc networks are self-organizing and self-re-configuring multi-hop wireless networks where, the structure of the network changes dynamically. This is mainly due to the mobility of the nodes. Nodes in these networks utilize the same random access wireless channel, cooperating in a friendly manner to engaging themselves in multi-hop forwarding. The nodes in the network not only act as hosts but also as routers that route data to/from other nodes in network.

A routing procedure is always needed to find a path so as to forward the packets appropriately between the source and the destination. Within a cell, a base station can reach all mobile nodes without routing via broadcast in common wireless networks. In the case of ad-hoc networks, each node must be able to forward data for other nodes.

4.1.2 Multipath Data Transmission

The application of multipath techniques in mobile ad hoc networks seems natural, as multipath routing allows diminishing the effect of unreliable wireless links and the constantly changing topology. The on-demand multipath routing scheme is presented in as a multipath extension of dynamic source routing (DSR) in which alternate routes are maintained, so that they can be utilized when the primary one fails. Another extension of DSR, multiple sources routing (MSR), proposes a weighted round-robin heuristic-based scheduling strategy among multiple paths in order to distribute load, but provides no analytical modeling of its performance. The split multipath routing (SMR), focuses on building and maintaining maximally disjoint paths, however, the load is distributed only in two routes per session.

4.2 Secure Data Transmission

Networks of thousands tiny sensor devices, which have low processing power, limited memory and energy, play roles for an economical solution to some challenging problems, such as, traffic monitoring, building safety, border security, habitat monitoring, tsunami alarm, medical emergency response and so on.

Undoubtedly security is an integral part of these applications. Authenticity of message is more important than confidentiality of data in this case. Consequently, if application does not consider adequate security measure then the intruder could find possible backdoor to feed highly abnormal information into the sensing devices and gain advantage of its own choice.

Secure data communication are done using the following techniques,

I. Secured Route Discovery by SMT

Secured routes are provided by establishing an End-to-End security association between the source and the destination. This scheme won't consider the intermediate nodes that may exhibit arbitrary and malicious behavior. The source node S and destination node T negotiate a shared secret key- KS, T with the knowledge of each other's public key.

II. Security Provided by SMT under Various Attacks

1) Fake Reply

If M1 receives the request by S and reply a fake route to S, that false reply will be discarded by the source since M1 doesn't know KS,T and not able to produce a valid MAC.

2) Tampering Route Reply

If the malicious nodes M1 or M2 changes the route reply send by T, S will discard it as the modified reply won't integrate with the expected MAC of T.

3) Resource Consumption Attack

If the adversaries want to exhaust the network resources then they will replay the requests. On receiving the replayed requests, the nodes will drop the requests based on query identifiers.

4) Fabricated Route Requests

Malicious nodes after observing for some time the requests generated by source it will fabricate several queries with subsequent query identifiers. The goal is the intermediate nodes will store this numbers and drop out the legitimate requests sent by the source. This type of attack cannot be prevented by SMT.

5) Spoofing Attack

The nodes M1 and M2 may spoof an IP address and participate in the route requests. This attack cannot be identified and they can hide their location by masking.

6) Colluding nodes Attack

If the nodes colluded during both the request and reply phase, the source will accept the false route information. For example in M1 tunnels the route request to M2.M2 will broadcast the route request with route segment between M1 and M2 falsified. In the reverse direction, T will consider this path and send the route reply back to the source through M2.Reply is reverse tunneled by M2 to M1.By this a false path will be included between S and T.

III. Secured Data Communication of SMT

1. Active Path Sets(APS) and Message Transmission

A set of diverse, node disjoint multiple paths are selected by applying secured route discovery protocol. The set of paths used for current data transmission are known as Active Path Sets. The message is dispersed based on Robin's algorithm and is transmitted in multiple paths by dispersing it into pieces and after encoding. Redundancy ensures successful reconstruction of data even if some loss occurs due to malicious nodes or breakage of routes.

2. Robust Feedback Mechanism

Each dispersed piece is transmitted in different route and carries a Message Authentication Code and by that the integrity of the message and authenticity of the source is verified. After validation, the destination acknowledges every successful receipt. The feedback mechanism is also cryptographically protected and dispersed.

3. APS Adaptation

Successful receipt of ACKS indicates operational routes while missing ACK implies that the route is either broken or compromised. The paths are rated based on short term and long term rating. The routes are selected or discarded based on their rates.

5 IMPLEMENTATION

System implementation is a stage in the project where the theoretical designs turned into working system. The most crucial stage is the user confidence that the new system will work effectively and efficiently.

The performance of reliability of the system was tested and it gained acceptance. The system was implemented successfully. Implementation is a process that means converting a new system into operation.

Proper implementation is essential to provide a reliable system to meet organization requirements. During the implementation stage a live demon was undertaken and made in front of end-users. Implementation is a stage of project when the system design is turned into a working system.

The stage consists of the following steps.

- Testing the developed program with sample data.
- Detection and correction of internal error.
- Testing the system to meet the user requirement.
- Feeding the real time data and retesting.
- Making necessary change as described by the user.

5.1 Input Design

Input Design is the part of overall system design that requires very careful attention. If the data going into the system is incorrect then the processing and the output will affect by these errors.

The inputs in the system are of three types:

- External : Prime inputs for the system
- Internal : User communication with the system
- Interactive : Inputs entered during a dialog with the computer

The above input types enrich the proposed system with numerous facilities that make it more advantageous in comparison with the exiting normal system. All the inputs entered are completely raw, initially, before being entered into a database, each of them available processing.

5.2 Output Design

Intelligent output design will improve systems relationships with the user and help in decision making. Outputs are also used to provide a permanent hardcopy of the results for latter consultations. The most important reason, which tempts the user to go for a new system is the output. The output generated by the system is often regarded as the criterion for evaluating the usefulness for the system.

Here the output requirements use to be predetermined before going to the actual system design.

The output design is based on the following:

- Determining the various outputs to be presented to the user.
- Differentiating between inputs to be displayed and those to be printed.
- The format for the presentation of the outputs.

5.3 NS2 CODE

TCL Script to Create Wireless Ad-hoc Network in NS2 Environment

```
set val(chan)    Channel/WirelessChannel     ;# channel type
set val(prop)    Propagation/TwoRayGround    ;# radio-propagation model
set val(netif)   Phy/WirelessPhy/802_15_4    ;# network interface type
set val(mac)     Mac/802_15_4                ;# MAC type
set val(ifq)     Queue/DropTail/PriQueue     ;# interface queue type
set val(ll)      LL                          ;# link layer type
set val(ant)     Antenna/OmniAntenna         ;# antenna model
set val(ifqlen)  60                          ;# max packet in ifq
set val(nn)      30                          ;# number of mobilenodes
set val(rp)      AODV                        ;# routing protocol
set val(a)       100                         ;#
set val(b)       100                         ;#
set val(nam)     out.nam                     ;#
set val(traffic) ftp                         ;#

proc getCmdArgu {argc argv} {
      global val
      for {set c 0} {$c < $argc} {incr c} {
            set arg [lindex $argv $c]
            if {[string range $arg 0 0] != "-"} continue
                set name [string range $arg 1 end]
                set val($name) [lindex $argv [expr $c+1]]
      }
}
getCmdArgu $argc $argv
set applTime1     0.1     ;
set applTime2     0.4     ;
set applTime3     0.8     ;
set applTime4     12      ;
set sTime         16      ;

#---------------Initialize Global Variables-----------------#
set ns_              [new Simulator]

# Creating trace file and nam file
set tracefd    [open out.tr w]
$ns_ trace-all $tracefd
if { "$val(nam)" == "out.nam" } {
      set namtrace    [open ./$val(nam) w]
      $ns_ namtrace-all-wireless $namtrace $val(a) $val(b)
}
$ns_ puts-nam-traceall {# nam4wpan #}   ;# inform to nam that this is a trace file
for wpan (special handling needed)
Mac/802_15_4 wpanNam namStatus on          ;# default = off (should be turned
on before other 'wpanNam' commands can work)

# For model 'TwoRayGround'
set dstnce(4m)  7.61113e-06
set dstnce(8m)  2.34381e-06
set dstnce(11m) 1.94278e-06
set dstnce(12m) 1.56908e-06
set dstnce(13m) 1.38527e-06
set dstnce(14m) 1.23774e-06
set dstnce(15m) 9.86011e-07
set dstnce(16m) 8.54570e-07
set dstnce(18m) 7.51287e-07
set dstnce(21m) 4.80696e-07
set dstnce(26m) 3.07845e-07
set dstnce(30m) 2.13343e-07
set dstnce(36m) 1.56262e-07
set dstnce(42m) 1.20574e-07
Phy/WirelessPhy set CSThresh_ $dstnce(16m)
Phy/WirelessPhy set RXThresh_ $dstnce(16m)

# set up topography object
set topo      [new Topography]
$topo load_flatgrid $val(a) $val(b)

# Create God
set god_ [create-god $val(nn)]
set chan_1_ [new $val(chan)]
# configure node
$ns_ node-config -adhocRouting $val(rp) \
                 -llType $val(ll) \
                 -macType $val(mac) \
                 -ifqType $val(ifq) \
                 -ifqLen $val(ifqlen) \
                 -antType $val(ant) \
                              -propType $val(prop) \
                              -phyType $val(netif) \
                              -topoInstance $topo \
                              -agentTrace ON \
                              -routerTrace OFF \
                              -macTrace ON \
                              -movementTrace OFF \
                              -channel $chan_1_
for {set c 0} {$c < $val(nn) } {incr c} {
          set node_($c) [$ns_ node]
          $node_($c) random-motion 0                         ;
}
source Final.scn

# Setup traffic flow between nodes
proc cbrtraffic { srce dest interval strtTime } {
  global ns_ node_
  set udp($srce) [new Agent/UDP]
  eval $ns_ attach-agent \$node_($srce) \$udp($srce)
  set null($dest) [new Agent/Null]
  eval $ns_ attach-agent \$node_($dest) \$null($dest)
  set cbr($srce) [new Application/Traffic/CBR]
  eval \$cbr($srce) set packetSize_ 75
  eval \$cbr($srce) set interval_ $interval
  eval \$cbr($srce) set random_ 0
  #eval \$cbr($srce) set maxpkts_ 100000
  eval \$cbr($srce) attach-agent \$udp($srce)
  eval $ns_ connect \$udp($srce) \$null($dest)
  $ns_ at $strtTime "$cbr($srce) start"
 }
  proc poissontraffic { srce dest interval strtTime } {
  global ns_ node_
  set udp($srce) [new Agent/UDP]
  eval $ns_ attach-agent \$node_($srce) \$udp($srce)
  set null($dest) [new Agent/Null]
  eval $ns_ attach-agent \$node_($dest) \$null($dest)
  set expl($srce) [new Application/Traffic/Exponential]
  eval \$expl($srce) set packetSize_ 75
  eval \$expl($srce) set burst_time_ 0
  eval \$expl($srce) set idle_time_ [expr $interval*1000.0-70.0*8/260]   ;# idle_time
+   pkt_tx_time = interval
  eval \$expl($srce) set rate_ 260k
  eval \$expl($srce) attach-agent \$udp($srce)
  eval $ns_ connect \$udp($srce) \$null($dest)
  $ns_ at $strtTime "$expl($srce) start"
 }
  if { ("$val(traffic)" == "cbr") || ("$val(traffic)" == "poisson") } {
  puts "\nTraffic: $val(traffic)"

  #Mac/802_15_4 wpanCmd ack4data on
  puts [format "Acknowledgement for data: %s" [Mac/802_15_4 wpanCmd
ack4data]]
  set lspeed 0.6ms
  set hspeed 1.6ms
  Mac/802_15_4 wpanNam PlaybackRate $lspeed
  $ns_ at [expr $applTime1+0.1] "Mac/802_15_4 wpanNam PlaybackRate
$hspeed"
$ns_ at $applTime2 "Mac/802_15_4 wpanNam PlaybackRate $lspeed"
$ns_ at [expr $applTime2+0.1] "Mac/802_15_4 wpanNam PlaybackRate $hspeed"
$ns_ at $applTime3 "Mac/802_15_4 wpanNam PlaybackRate $lspeed"
$ns_ at [expr $applTime3+0.1] "Mac/802_15_4 wpanNam PlaybackRate $hspeed"
$ns_ at $applTime4 "Mac/802_15_4 wpanNam PlaybackRate $lspeed"
$ns_ at [expr $applTime3+0.1] "Mac/802_15_4 wpanNam PlaybackRate $hspeed"
eval $val(traffic)traffic 18 7 0.1 $applTime1
eval $val(traffic)traffic 11 5 0.1 $applTime2
eval $val(traffic)traffic 4 3 0.1 $applTime3
eval $val(traffic)traffic 29 26 0.1 $applTime4
Mac/802_15_4 wpanNam FlowClr -p AODV -c tomato
Mac/802_15_4 wpanNam FlowClr -p ARP -c aqua
if { "$val(traffic)" == "cbr" } {
```

```
          set packetType cbr
  } else {
          set packetType exp
  }
Mac/802_15_4 wpanNam FlowClr -p $packetType -s 18 -d 6 -c red
Mac/802_15_4 wpanNam FlowClr -p $packetType -s 11 -d 5 -c aqua4
Mac/802_15_4 wpanNam FlowClr -p $packetType -s 4 -d 3 -c cyan4
Mac/802_15_4 wpanNam FlowClr -p $packetType -s 29 -d 26 -c yellow4
$ns_ at $applTime1 "$node_(18) NodeClr red"
$ns_ at $applTime1 "$node_(7) NodeClr red"
$ns_ at $applTime1 "$ns_ trace-annotate \"(at $applTime1) $val(traffic)
Identifing defects between sensors 18 and node 7\""
$ns_ at $applTime2 "$node_(11) NodeClr aqua4"
$ns_ at $applTime2 "$node_(5) NodeClr aqua4"
$ns_ at $applTime2 "$ns_ trace-annotate \"(at $applTime2) $val(traffic)
Identifing defects between sensors 11 and node 5\""
$ns_ at $applTime3 "$node_(4) NodeClr cyan3"
$ns_ at $applTime3 "$node_(3) NodeClr cyan3"
$ns_ at $applTime3 "$ns_ trace-annotate \"(at $applTime3) $val(traffic)
Identifing defects between sensors 4 and node 3\""
$ns_ at $applTime4 "$node_(29) NodeClr yellow4"
$ns_ at $applTime4 "$node_(26) NodeClr yellow4"
$ns_ at $applTime4 "$ns_ trace-annotate \"(at $applTime1) $val(traffic)
Identifing defects between sensors 29 and node 26\""
}
proc ftptraffic { srce dest strtTime } {
global ns_ node_
set tcp($srce) [new Agent/TCP]
eval \$tcp($srce) set packetSize_ 60
set sink($dest) [new Agent/TCPSink]
eval $ns_ attach-agent \$node_($srce) \$tcp($srce)
eval $ns_ attach-agent \$node_($dest) \$sink($dest)
eval $ns_ connect \$tcp($srce) \$sink($dest)
set ftp($srce) [new Application/FTP]
eval \$ftp($srce) attach-agent \$tcp($srce)
$ns_ at $strtTime "$ftp($srce) start"
}
if { "$val(traffic)" == "ftp" } {
  puts "\nTraffic: ftp"
  puts [format "Acknowledgement for data: %s" [Mac/802_15_4 wpanCmd
ack4data]]
  set lspeed 00.25ms
  set hspeed 01.6ms
  Mac/802_15_4 wpanNam PlaybackRate $lspeed
  $ns_ at [expr $applTime1+0.1] "Mac/802_15_4 wpanNam PlaybackRate
$hspeed"
  $ns_ at $applTime2 "Mac/802_15_4 wpanNam PlaybackRate $lspeed"
  $ns_ at [expr $applTime2+0.1] "Mac/802_15_4 wpanNam PlaybackRate
$hspeed"
  $ns_ at $applTime3 "Mac/802_15_4 wpanNam PlaybackRate $lspeed"
  $ns_ at [expr $applTime3+0.1] "Mac/802_15_4 wpanNam PlaybackRate 6ms"
  $ns_ at $applTime4 "Mac/802_15_4 wpanNam PlaybackRate $lspeed"
  $ns_ at [expr $applTime3+0.1] "Mac/802_15_4 wpanNam PlaybackRate 11ms"
  ftptraffic 18 7 $applTime1
  ftptraffic 11 5 $applTime2
  ftptraffic 4 3 $applTime3
  ftptraffic 29 26 $applTime4
  Mac/802_15_4 wpanNam FlowClr -p AODV -c tomato
  Mac/802_15_4 wpanNam FlowClr -p ARP -c aqua
Mac/802_15_4 wpanNam FlowClr -p tcp -s 18 -d 7 -c red
Mac/802_15_4 wpanNam FlowClr -p ack -s 7 -d 18 -c red
Mac/802_15_4 wpanNam FlowClr -p tcp -s 11 -d 5 -c aqua4
Mac/802_15_4 wpanNam FlowClr -p ack -s 5 -d 11 -c aqua4
Mac/802_15_4 wpanNam FlowClr -p tcp -s 4 -d 3 -c cyan4
Mac/802_15_4 wpanNam FlowClr -p ack -s 3 -d 4 -c cyan4
Mac/802_15_4 wpanNam FlowClr -p tcp -s 29 -d 26 -c yellow4
Mac/802_15_4 wpanNam FlowClr -p ack -s 26 -d 29 -c yellow4
$ns_ at $applTime1 "$node_(18) NodeClr red"
$ns_ at $applTime1 "$node_(7) NodeClr red"
$ns_ at $applTime1 "$ns_ trace-annotate \"(at $applTime1)  Identifing defects
```

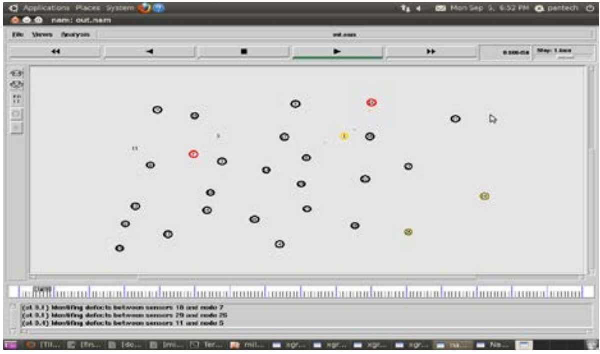
```
between sensors 18 and node 7\""
$ns_ at $applTime2 "$node_(11) NodeClr aqua4"
$ns_ at $applTime2 "$node_(5) NodeClr aqua4"
$ns_ at $applTime2 "$ns_ trace-annotate \"(at $applTime2) Identifing defects
between sensors 11 and node 5\""
$ns_ at $applTime3 "$node_(4) NodeClr cyan3"
$ns_ at $applTime3 "$node_(3) NodeClr cyan3"
$ns_ at $applTime3 "$ns_ trace-annotate \"(at $applTime3) Identifing defects
between sensors 4 and node 3\""
$ns_ at $applTime1 "$node_(29) NodeClr yellow4"
$ns_ at $applTime1 "$node_(26) NodeClr yellow4"
$ns_ at $applTime1 "$ns_ trace-annotate \"(at $applTime1)  Identifing defects
between sensors 29 and node 26\""
 }

# defines the node size in nam
for {set c 0} {$c < $val(nn)} {incr c} {
                $ns_ initial_node_pos $node_($c) 2
}

# Tell nodes when the simulation ends
for {set c 0} {$c < $val(nn) } {incr c} {
   $ns_ at $sTime "$node_($c) reset";
}
$ns_ at $sTime "stop"
$ns_ at $sTime "puts \"\nNS EXITING...\""
$ns_ at $sTime "$ns_ halt"
proc stop {} {
   global ns_ tracefd val env
   $ns_ flush-trace
   close $tracefd
   set hasDISPLAY 0
   foreach index [array names env] {
      #puts "$index: $env($index)"
      if { ("$index" == "DISPLAY") && ("$env($index)" != "") } {
         set hasDISPLAY 1
      }
   }
   if { ("$val(nam)" == "out.nam") && ("$hasDISPLAY" == "1") } {
   exec rm -f out-tcp.xgr
   exec awk -f avg_throughput.awk out.tr > BANDWIDTH
   exec xgraph BANDWIDTH &
   exec rm -f out-tcp.xgr
   exec awk -f graph2.awk out.tr > DELAY
   exec xgraph DELAY &
   exec rm -f out-tcp.xgr
   exec awk -f graph3.awk out.tr > LIFETIME
   exec xgraph LIFETIME &
   exec nam out.nam &
   }
}
puts "\nStarting Simulation..."
$ns_ run
```

6 SYSTEM TESTING

6.1 Results and Discussions

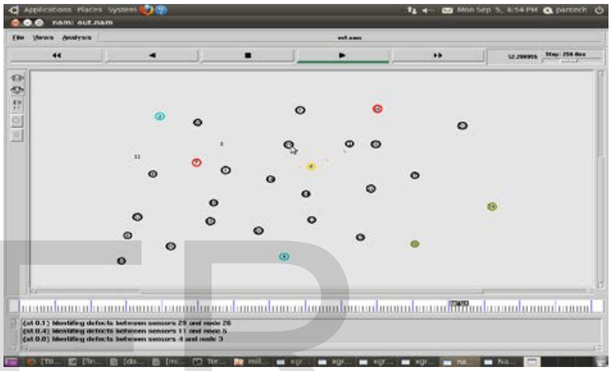
6.1.1 Nam

NAM provides a visual interpretation of the network topology created. Nam can be executed directly from Tcl script and it presents information such as throughput, number packets on each link.



```
(at 0.1) Identifing defects between sensors 18 and node 7
(at 0.1) Identifing defects between sensors 29 and node 26
(at 0.4) Identifing defects between sensors 11 and node 5
```

Fig. 1. AODV Without Malicious



```
(at 0.1) Identifing defects between sensors 29 and node 26
(at 0.4) Identifing defects between sensors 11 and node 5
(at 0.8) Identifing defects between sensors 4 and node 3
```

Fig. 2 AODV With Malicious

6.1.2 X-Graph

X-Graph is an X-Windows application that includes:

- Interactive plotting and graphing
- Animation and derivatives

To use X-Graph in NS2 the executable can be called within a tcl script. Then
It loads a graph displaying the information visually of the trace file produced from the simulation.

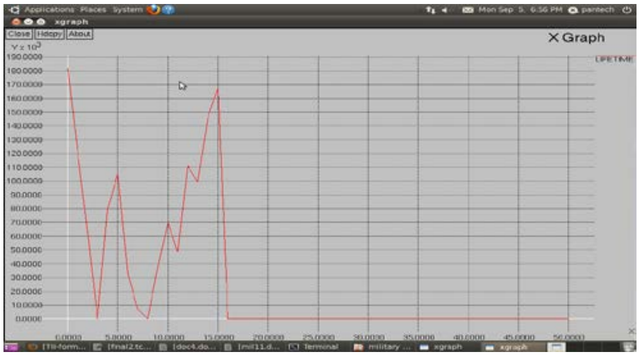


Fig. 3. AODV- Bandwidth

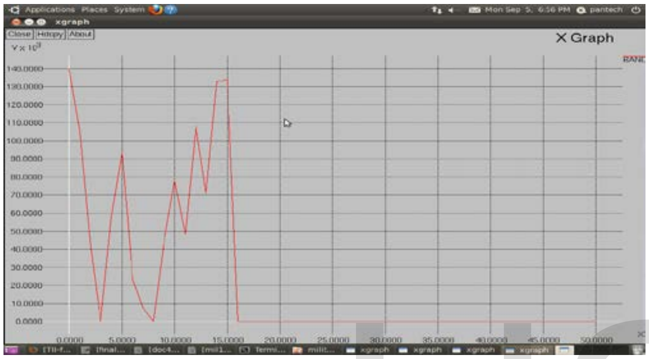


Fig. 4. AODV-Delay

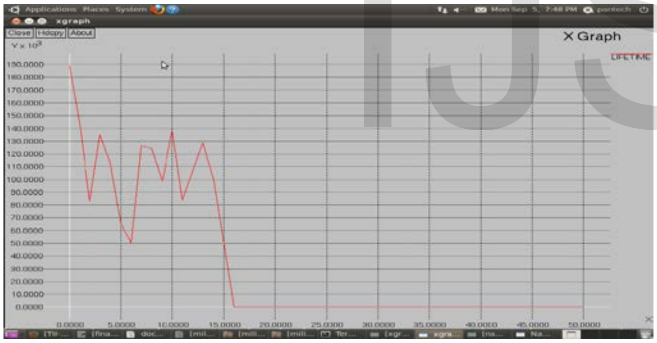


Fig. 5. AODV- Lifetime

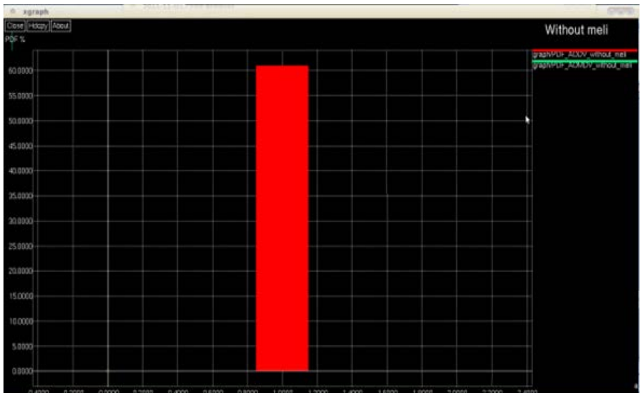


Fig. 6. Without malicious environment

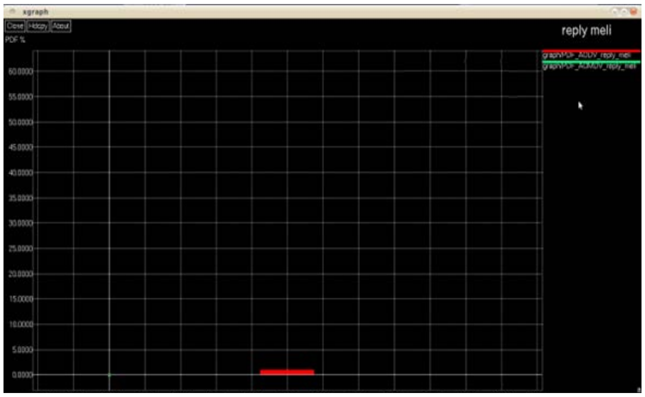


Fig. 7. With reply malicious

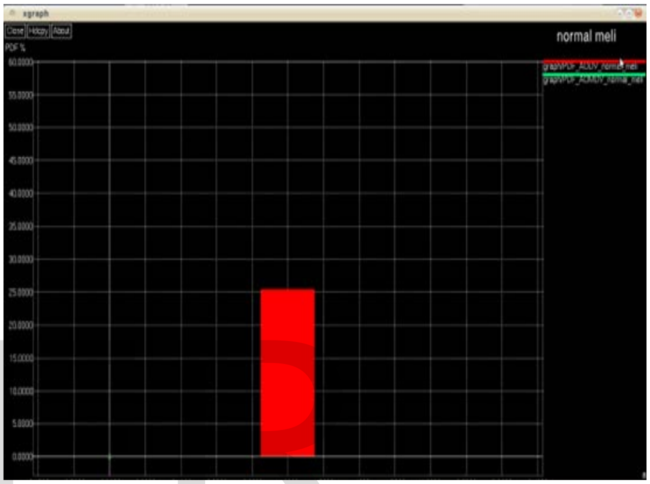


Fig. 8. With normal malicious

7. CONCLUSION

Simulations were conducted using the NS2 network simulator. Nodes in the network were configured to use 802.11 radios with a bandwidth of 2 Mbps and a nominal range of 250 m. It was not possible to simulate the flood rushing attack using this technique because it requires timing changes in the routing protocol code. The module is implemented as part of the NS2 Link Layer (LL) object which lies directly below the Routing Agent and directly above the MAC layer.

In our project we are using the popular simulation tool, which name as NS2. In our simulation tool, we can show the two different types of outputs such as Nam (Network animator) window and X-graph. In our network all wireless nodes are deployed in random manner. And we are creating one source and destination.

According to our proposed concept we are providing multipath communication for security purpose.

In our network model, we are starting the communication after 14th second so there is no communication before 14th second. In the period of 14-22 second we are providing the single path communication, so source node select only the single path communication for transferring the data to destination, this is existing model output. Here single path communication mean transmission path should node been changed other than path or node failure. So in this type of communication may cause to hacking in network.

To avoid data hacking in our network we are going to implement multipath communication in our network. Here we are considering our communication path is changeable even path or node is node failed. So data is sending through different paths, it provide high security than single path, this result is shown in same Nam window in the period of 23-30 second.

In above discussion, we are implemented secure message transmission in MANET. This model is proving security but not QoS, so we have to improve this model to provide high quality of service. Due to need of quality of service we are going to implement energy efficient transmission in our network model, due to some technical reason we can't so that QoS.

Our transmission time is started after 14th sec only so there is no y axis value before 14th second. The AODV model communication duration is from 14 to 22 second.

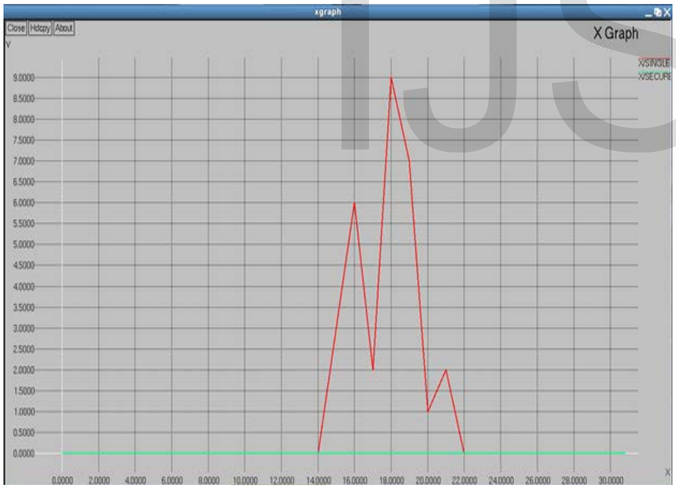


Fig. 9. Graph 1

REFERENCES

[1] A. Tsirigos, Z. J. Hass (2004). "Analysis of multi path routing, Part 1: The effects on the packet delivery ratio" IEEE Transactions on Wireless Communication., vol.3, no.2, pp: 500- 511.

[2] Banner, R. Orda, A, "Multipath Routing Algorithms for Congestion Minimization". This paper appears in: Networking, IEEE/ACM Transactions on Publication Date: April 2007 Volume: 15, Issue: 2, on page(s): 413-424.

[3] J. Peng, B. Sikdar, L. Cheng(2009) "Multicasting with Localized Control in Wireless Ad Hoc Networks" IEEE Transaction on Mobile Computing.

[4] S. A. Ahmadi Olounabadi, A. Damodaram, V. K. Prasad, M. Hosseini (2016). "Impact of Multi-Path Security in Wireless Ad Hoc Networks in Indoor Environments by using AOMDV Methods". International Journal of Engineering and Advanced Technology (IJEAT). ISSN: 2249 – 8958, Volume-6, Issue-2, December 2016. Page(s): 24-36.

[5] Y. Zhu, X. Fu, B. Graham, R. Bettati, W. d. Zhao (2004). "On Flow Correlation Attacks and Countermeasures in combine Networks".

[6] S. Capkun, L. sandwich, J.P. Hubaux (2003). "Self-Organized Public-Key Management for Mobile impromptu Networks".

[7] J. Kong, X. Hong (2003). "ANODR: Anonymous on Demand Routing with untraceable Routes for Mobile Ad-hoc Networks".

[8] B. Zhu, Z. Wan, M. S. Kankanhalli, F. Bao, R. H. Deng (2004). "Anonymous Secure Routing in Mobile Ad-Hoc Networks".

[9] S. Seys, A. Preneel (2009). "ARM: Anonymous Routing Protocol for Mobile impromptu Networks".

[10] S. Capkun, L. Buttyan, and J. Hubaux (2003). "Self-organized public-key management for mobile ad hoc networks," IEEE Trans. Mobile Compute., vol. 2, no. 1, pp. 52–64, Jan.-Mar. 2003.

[11] J. Kong, X. Hong. "ANODR: anonymous on demand routing with untraceable routes for mobile ad-hoc networks," in Proc. ACM MOBIHOC', 03, pp. 291–302.

[12] B. Zhu, Z. Wan, F. Bao, R. H. Deng, M. K. Halli (2004). "Anonymous secure routing in mobile ad-hoc networks," in Proc. 2004 IEEE Conference on Local Computer Networks, pp. 102–108.

[13] S. Seys, B. Preneel (2006). "ARM: anonymous routing protocol for mobile ad hoc networks," in Proc. 2006 IEEE International Conference on Advanced Information Networking and Applications, pp. 133–137.

[14] L. Song, L. Korba, and G. Yee. "AnonDSR: efficient anonymous dynamic source routing for mobile ad-hoc networks," in Proc. 2005 ACM Workshop on Security of Ad Hoc and Sensor Networks, pp. 33–42.

[15] Y. Dong, T. W. Chim, V. O. K. Li, S.-M. Yiu, C. K. Hui (2009). "ARMR: anonymous routing

protocol with multiple routes for communications in mobile ad hoc networks," Ad Hoc Networks, vol. 7, no. 8, pp. 1536–1550, 2009.

AUTHOR'S PROFILE

I. Seyed Amin Ahmadi Olounabadi, Ph.D. scholar student in Computer Science and Engineering, Dept. of Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad, Telangana , India, ,Research interest: Network and Network Security, IT, Network Management.

II. Prof. Avula. Damodaram, Vice-Chancellor Sri Venkateswara University, Tirupati, Andhra Pradesh, India. Faculty of Computer Science & Engineering at JNTU, Hyderabad, Research interests: include Image Processing, Pattern Recognition, Network Security, Steganography and Digital Watermarking.

III. Prof. V Kamakshi Prasad, Director of Evaluation and Professor of Computer Science and Engineering Department, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad, Telangana, India. Research interests: Speech Recognition and Processing, Image processing, Pattern Recognition, Data Mining, Ad-hoc networks, Computer Graphics.