Computers, Materials & Continua
DOI: 10.32604/cmc.2022.026141
Article

Tech Science Press

Resistance to Malicious Packet Droppers Through Enhanced AODV in a MANET

Shirina Samreen*

Department of Computer Science, College of Computer and Information Sciences, Majmaah University, Al Majmaah, 11952, Saudi Arabia
*Corresponding Author: Shirina Samreen. Email: s.samreen@mu.edu.sa
Received: 16 December 2021; Accepted: 22 February 2022

Abstract: Packet dropping in a mobile ad hoc network can manifest itself as the data plane attacks as well as control plane attacks. The former deal with malicious nodes performing packet drop on the data packets following the route formation and the latter deal with those malicious nodes which either drop or manipulate the control packets to degrade the network performance. The idea of the proposed approach is that during the route establishment, each of the on-path nodes is provided with pre-computed hash values which have to be used to provide a unique acknowledgement value to the upstream neighbor which acts as a proof of the forwarding activity. The analysis phase results in the detection of nodes which exhibited malicious behavior in the current communication session so as to avoid them in the future communication sessions resulting in an improved packet delivery fraction even in the presence of one or more malicious nodes in the network. The communication overhead incurred is minimum since the acknowledgement reports are sent to the destination for a transmission of N packets rather than an individual acknowledgement for each transmitted packet. In contrast to some of the existing techniques, the proposed mechanism is not dependent on the deployment of additional infrastructure like special Intrusion Detection System (IDS) nodes. The only overhead incurred is in the form of control packets exchanged for the reports request and the reports submission.

Keywords: Mobile ad hoc networks (MANETs); routing protocols; attacks on MANET; secure routing; packet droppers

1 Introduction

Numerous security compromise issues are associated with a MANET in the route formation as well as the data transmission phase. Most of these can be attributed to the inherent features like open communication medium, on-the-fly route formation, frequent route changes due to mobility-based link breaks. Malicious packet dropping launched after the route formation is quite challenging to detect and can have deterring effects on the overall throughput. The biggest challenge is to distinguish between malicious and non-malicious packet dropping. Non-malicious packet dropping can happen

because of a number of reasons listed as follows: attempt to preserve the battery power by a relay node, congestion in the network and noise within the communication channel. Malicious packet dropping is associated with the scenario when an attacker captures a node and has full control on it causing it to drop packets deteriorating the overall performance of the network.

The ***aim*** of the research is as follows: a security mechanism is proposed that can act as an add-on to any reactive routing protocol such that packet delivery fraction is maintained at a modest level even in the presence of multiple adversarial nodes dropping the data packets. The research ***contribution*** involves a security enhancement to AODV routing protocol as follows: Each on-path node has an acknowledgement report comprising a proof of all the packets it received as well as the ones it forwarded which is always verifiable by the source node. The security mechanism assumes that the on-path nodes participating in a communication session are always trusted and responsible for detecting the adversarial nodes. The overhead incurred is in the form of control packets exchanged for the request and collection of reports from each of the intermediate nodes for each session to perform the report analysis so as to identify the misbehaving nodes dropping the data packets in order to eliminate them from participating in the next successive session. The ***motivation*** of the proposed research is as follows: Unlike other techniques to counter the selective black hole attacks like [1], the proposed mechanism does not need the deployment of any special IDS (intrusion detection system) nodes and the control overhead is minimized. It is done by appropriately choosing the number of packets to be transmitted for each session and is justifiable with the maintenance of modest packet delivery fraction even in the presence of multiple adversaries.

The manuscript content is organized as follows: Section 1 describes the introduction. Following is Section 2 which describes related work describing various defense approaches to protect against blackhole nodes. Section 3 explains the modifications/enhancements to the baseline routing protocol Ad-hoc On-demand Distance Vector (AODV). Section 4 explains the proposed security scheme followed by Section 5 enlisting the optimizations to proposed approach. Section 6 presents the performance analysis followed by section describing the conclusion and future work.

2 Related Work

In [2], an approach has been proposed which is based upon probing wherein the source node randomly selects an intermediate node on the source to destination path which has to send a report to the destination appending it to the data packet which it forwards to the next hop neighbour. The destination over a period of time receives the reports from all the intermediate nodes and analyses them from traffic deviations. These are secured from manipulation by other malicious nodes. It is achieved through the use of a chain of Hash-based Message Authentication Code (HMAC)s on link layer acknowledgements. Symmetric cryptographic construction is used to ensure the secrecy of random node selected for sending the reports. The protocol is associated with a small communication overhead due to the increase in packet size caused by appending the reports with data packets.

Detection of malicious packet dropper nodes based upon the ideas from opportunistic networking was proposed in [3]. The mechanism is based upon the usage of two types of nodes known as Cop node and a regular node. The Cop node is solely responsible for the nodes which perform packet dropping either with malicious intent or selfish nature whereas the regular nodes only perform normal data transmission without worrying about the detection of packet droppers. The idea is to reduce the burden of detection of misbehavior upon regular nodes at the cost of deploying additional nodes which are dedicated solely for misbehavior detection. A threshold-based approach with Dynamic source routing protocol is used wherein the cop node operates in the promiscuous mode and keeps moving within

the wireless channel in an attempt to overhear the ongoing communication. It keeps monitoring the forwarding behavior of its neighboring regular nodes and when the number of packets dropped exceeds a threshold, it sends an alarm message to the source which results in excluding the packet droppers from source to destination path by generating an alternate route.

A security model was proposed employing a reputation management [4] having certain nodes as mangers and the rest as common nodes. The former nodes are responsible for the collection of information followed by the trust computation reflecting the packet forwarding behavior of the latter nodes. The latter nodes monitor their peers and send the information managers. The honesty of manager nodes decides the efficiency of the proposed research.

A novel security model employing a metric termed as blacklist threshold representing the quantity of amount of node usage within the network involving the node's contribution towards packet forwarding was proposed [5]. It was named as Trust Model focusing on node Usage against MANETs On-Off Attack (TMUMO).

A security approach measuring the positive packet forwarding behavior based upon historical data was proposed [6]. The method is having comprehensive computation method for calculating the metric through direct as well as indirect observations but it is not resilient to attackers attempting to corrupt the computation process.

Yet another latest research [7] creates a trust model for a MANET in the context of Internet of Things (IoT). Subsequently, it involves the design of a routing protocol to guard against the packet droppers. Apart from the design of security schemes, other related topics including authentication, secure routing and minimization of overhead have been reviewed in detail [8].

In a MANET, the malicious packet droppers tend to modify their behavior periodically so as to evade detection and continue with the malicious activity. A totally decentralized mechanism was proposed [9] to monitor the node's behavior and detect the malicious activity despite the change in their behavior. It is based upon two models for the classification and tracking of the evolving behavior namely Bernoulli Bayesian model and Markov Chain model respectively.

The detection of the most vulnerable form of packet dropping known as black hole attack in the form of an intrusion detection system was proposed [10] termed as 'Accurate and Cognitive Intrusion Detection System' (ACIDS). It works by considering the deviation of the parameters like Sequence number of the destination node and route reply.

A novel security mechanism to protect against selective packet dropping attack was proposed [11] named as Resistive to Selective Drop Attack (RSDA) scheme. It can be unified with the existing routing protocols like AODV and Dynamic Source Routing (DSR) to provide security. The working mechanism is based upon deactivating the highest weight link and providing the authentication through the elliptic curve digital signature algorithm.

A secure routing protocol to detect packet drop attacks has been proposed [12] which discovers the best route from source to destination based upon priority assigned to a number of parameters like the hop-count, the battery life, and security. The most important feature of the protocol is the application of energy-saving scheme to conserve the energy. The protocol is named as Ad Hoc On-Demand and Distance Vector–Packet Drop Battling Mechanism (AODV–PDBM).

An approach to distinguish between malicious and non-malicious packet dropping was proposed [13] wherein the concept of node trust is employed. It reflects the behavioral traits of a node while participating in the packet forwarding after the route establishment. Another recent research to deal with packet droppers using trust and reputation management [14] has the nodes in a wireless network

rate each other and use the aggregate ratings to compute the trust. A data mining approach known as fuzzy based secure architecture was proposed [15] to classify the nodes using packet delivery ratio, forwarding and residual energy as input parameters.

The problem of blackhole attacks is addressed to have a secure transmission of data through a new method termed as modified Ad-hoc On-demand Multipath Distance Vector (AOMDV) protocol [16]. The data transmission is done by computing multiple routes to reach the destination employing the homomorphic encryption method.

In a MANET environment, the source of packet dropping is either malicious nodes or link error. To detect malicious packet droppers, a novel mechanism is proposed known as the improved failure aware third party Auditor (IFTPA) [17] based homomorphism linear authenticator (HLA) mechanism (IFHM) with the Secured Ad-hoc On-demand Distance Vector (SAODV). It involves a mechanism to revoke a server as well as reassign a server making the security scheme robust to the compromise of the server node. Hence the detection of malicious packet dropper can be done along with an appropriate authentication service.

A novel approach employing a trust metric comprising three dimensions (trust, distrust and uncertainty) to handle the dynamic MANET topology is termed as Enhanced Average Encounter Rate-AODV (EAER-AODV) protocol [18]. The trust model facilitates the secure route discovery by having the source node choose the next hop based upon the mobility of the nodes and the malicious nature to avoid any packet dropping either with non-malicious intent through link break or malicious intent through compromised nodes.

An optimization based trust-aware secure routing protocol was proposed termed as Atom Whale Optimization Algorithm (AWOA) [19]. Various factors reflecting the trust including the average rate of encounter, and frequency of cooperation, integrity factor, and the rate of packet forwarding are used to select the optimal route. The modelling of fitness function is performed by involving factors like mobility and trust to provide improved network performance despite the presence of malicious packet droppers.

Secure routing protocol employing the Ad-hoc On-Demand Multipath Distance Vector (AOMDV) as a baseline termed as Trusted Security Ad-hoc On-Demand Multipath Distance Vector (TS-AOMDV) was proposed [20] to detect and eliminate malicious nodes launching flooding, blackhole, and gray hole attacks in the network. The phases of route discovery and data transmission are involved in the attacks detection through an intrusion detection system (IDS). The IDS is incorporated in each node which measures a parameter within the Route Request (RREQ) packet during route discovery and estimates the rate of packet forwarding by the neighbors during the forwarding phase. The attack detection is done through a comparison against a threshold value followed by the computation of trust rates of a node to form a reliable route.

The proposed security mechanism in this paper, compared to the above mentioned mechanisms is neither dependent upon any special IDS nodes nor does it employ promiscuous neighbourhood monitoring. It employs a report based mechanism wherein each intermediate node maintains a track of its own forwarding activity. The reports are sent to the source node in the form of cumulative acknowledgement reports for each session (comprising of a fixed no. of data packets) rather than having an individual acknowledgement for each data packet, thereby maintaining a good packet delivery fraction even in the presence of adversarial nodes with a reduced overhead.

3 Modifications to AODV Routing Protocol

When the proposed security mechanism has to be used with AODV routing protocol [21], then the feature of having an intermediate node respond to an RREQ packet with a corresponding Route Rely (RREP) packet has to eliminated since the security mechanism has the source node create an RREQ packet embedded with an encrypted random value which has to be decrypted by the destination node for subsequent security related operations before sending the RREP packet. Hence the packet header format of the RREQ packet is modified to incorporate the encrypted random value and also accommodate the path information about the addresses of the intermediate nodes through which the RREQ packet has traversed.

When the RREQ packet reaches the destination, it utilizes this information from the encrypted random value to generate a set of n (where n is the total no. of intermediate nodes on the source to destination path) pre-computed hash values obtained by applying a secure hash function to the random value embedded within the RREQ packet. These n values are encrypted with the symmetric key shared by the destination with each of the on-path nodes and disseminated along the path. The n encrypted values are embedded within the RREP packet which needs the modification of the RREP packet header format. The RREP packet header also incorporates the final path information so as to update the source node about each of the intermediate nodes upon the so formed path.

During the data transmission, if a link break is detected then the Route Error (RERR) packet is sent to the nodes in precursor list of the routing table as in the basic AODV protocol. Additionally, in the proposed security mechanism the data packet on transmission at which the link break was detected is inspected to look into the address of the source node at which the packet originated and also the sequence number of the packet so as to fill two additional fields within the RERR packet header namely: the srcpkt and the pktsno respectively. The above information embedded within the RERR packet is used by the source node to determine the sequence number x of the packet from where the current session's packets were lost so as to retransmit all the packets of the current communication session from the sequence number x to N in the next successive session.

At the beginning of each communication session, the source node backs up a copy of all the packets transmitted so as to use them for retransmission in the next successive session in case of the reception of an RERR packet due to a link break being detected during the transmission and packets being lost during the transmission. Apart from the modified RREQ, RREP and RERR packets, the proposed mechanism makes use of certain additional packets namely: the acknowledgement (ACK) packet, the Reports Request (RPTRQ) packet, the CLEAR packet, the REPORT packet and the MALI packet.

In each communication session, each intermediate node upon the reception of a data packet has to send an acknowledgement value computed using the sequence number of the received packet and the pre-computed hash value obtained from the destination at the time of route establishment through the RREP packet. The computed value acts as a proof for the upstream node's forwarding activity and it is sent in the form of ACK packets.

The end of a communication session happens upon the expiry of a timer set at the beginning of the session for a time period required for the transmission of N packets. During this wait period, an RERR packet may be received which updates the brksno field (from its default value of -1) of the source node to a value greater than zero which is the sequence number of the packet at which the link break occurred. If no RERR packet is received, then it is implied that no link break has occurred during the data transmission session and the route is intact. Hence the same route may be used, for sending the reports request and receiving the reports from each of the on-path nodes. A CLEAR packet

is used to inform all the on-path nodes that the data transmission has completed and the reports may be sent to the source along the same route. If an RERR packet is received during the wait time, then it indicates that a link break has been detected and the route is not intact. Hence the source node has to establish a fresh path to each of the on-path nodes so as to send the reports request and receive the reports back along the corresponding reverse paths. The RPTRQ packet is used by the source node to broadcast and form separate paths with each of the on-path nodes for the reception of REPORT packets which have an embedded acknowledgement report within them.

As an optimization measure for the proposed security mechanism, the destination node upon the reception of the RPTRQ packet, rather than sending a REPORT packet to the source node, sends out an RREP packet with an embedded report within it in order to utilize the broadcast of RPTRQ packet for the dual purpose of reports reception as well as an additional fresh route formation. The freshly formed route may be utilized for the next successive data transmission session. For this purpose, the packet header of the RREP packet is modified to incorporate two fields namely: rpt and rpts. The field rpt represents a boolean flag which indicates whether the RREP packet has an embedded report within it and the rpts field is the embedded acknowledgement report from the destination.

After the reception of reports from all the on-path nodes, the source node performs the reports analysis which provides the forwarding behavior characteristics of each of the intermediate nodes during the data transmission session to determine those nodes which can be categorized as either suspicious nodes or malicious nodes. The malicious nodes also called as the black hole nodes are the adversarial nodes which have to be avoided during the route establishment. This information is incorporated in the MALI packets which contain the addresses of all the black holes. The MALI packets are broadcast throughout the network so as to inform all the nodes about the black hole nodes.

4 Proposed Security Mechanism

4.1 Reports Request

The transmission of N packets by the source node is followed by setting a timer for a time period needed for sending the N packets. During the wait period, the reception of an RERR packet causes the source node to update its break sno from its default value of -1.

The expiry of the timer is followed by the source node entering into the Reports Request phase which is performed as follows:

Case 1: The source node has the default value of -1 in break sno field since no RERR packet has been received during the wait period. Under these circumstances, as the link is intact the same route can be employed to receive the acknowledgement reports from each of the on-path nodes including the destination node. A CLEAR packet used to request reports along the existing path is sent along the same route by the source. Each of the on-path nodes, upon receiving the CLEAR packet send back the REPORT packet on the reverse path of the CLEAR packet. The on-path nodes also forward the CLEAR packet to the next downstream neighbor on the source to destination path. Finally, when the CLEAR packet arrives at the destination, it sends back the REPORT packet and discards the CLEAR packet.

Case 2: The source node has the break sno value greater than zero since an RERR packet has been received because of a link break. Under these circumstances, it implies a link break which indicates that the existing route cannot be used. Consequently, a RPTRQ packet (Reports Request) is generated

by the source node which contains three significant fields namely nodeList, listLength and recvLength which are described as follows:

nodeList → contains the names and status flag for each of the on-path nodes. The status flag indicates whether acknowledgement report has been received from the node or not.

listLength → the number of on-path nodes.

recvLength → the number of on-path nodes from whom the acknowledgement reports have been received.

The broadcast of RPTRQ packet across the network is done similar to that of RREQ packet. The source node sets a timer to an approximate time interval required for the reports to be received from all of the intermediate nodes. When the timer expires, the source node checks whether the acknowledgment reports have not been received from any of the on-path nodes. If yes, the source node resends the RPTRQ packet with the modified nodeList fields by updating the status flags in the nodeList field of those intermediate nodes from where the reports have been received and resets the timer to an interval required for the reports to arrive from the remaining nodes which have not yet sent the reports. This process is repeated until the reports are not received from all of the intermediate nodes.

The processing of the RPTRQ packet is done in two parts: In the first part, it is processed in a similar way as the RREQ packet wherein the reception of the RPTRQ packet causes each of the nodes through which it traverses to update the routing table entries of reverse route. In the second part of processing the RPTRQ packet, each node which receives the RPTRQ packet checks whether its name is included in the nodeList. If yes, it checks whether its status flag is set to 0 or not and if the status flag is set 0, then the node first decrements the value of recvLength and sends it's acknowledgement report for the ongoing session in the form of REPORT packet along the reverse route through which it received the RPTRQ packet. If the recvLength value is greater than zero, it further broadcasts the RPTRQ packet. Otherwise it discards the RPTRQ packet.

The intermediate nodes maintain two lists known as alert list and suspicious list (initially empty) which are updated while sending the REPORT packet. The acknowledgement report is analyzed to check for the number of packets which have been forwarded to the downstream neighbor but the ACK packet has not been received for that packet. If the number of such packets are greater than 20% of the total number packets forwarded, then the downstream node is checked for existence in the suspicious list. If the downstream node already exists in the suspicious list, then it is included into the alert list. Otherwise, it is included into the suspicious list to check for its behavior in the future communication sessions.

The significance of the alert list is that, all the nodes included into it are those nodes which have repeatedly exhibited malicious intent to its upstream neighbor node by not sending the ACK packets. Hence, the upstream node avoids its participation in the routes with such a node as it's downstream neighbor since it may result in, the source node wrongly including the non-malicious node into the malicious list because of the repeated malicious behavior of its downstream neighbor.

The alert list is used in the Secure Route Establishment phase since each intermediate node uses its alert list to see if the RREP packet is received from a node included in its alert list. If yes, then the RREP packet is dropped, otherwise it is forwarded.

The source node at the end of a data transmission session performs the reports request to the on-path nodes and the on-path nodes submit the reports to the source node as follows: There are two cases to be considered: when no link break occurs and when a link break occurs in the data transmission

path. In the former case, the CLEAR packet unicasted by the source node using the same path results in the sending of REPORT packets by all the on-path nodes along the route in reverse direction. In the latter case, the link between nodes C and E is shown as broken which causes the source node to broadcast the RPTRQ packets followed by sending of REPORT packets by all the on-path nodes and the destination node along the reverse path.

4.2 Processing of Reports

The reception of REPORT packets by the source node results in the **Reports Processing** phase which composes two lists: ***suspicious list*** and ***malicious list*** (or blacklist). The former list comprises the identities of the nodes exhibiting misbehavior along with the frequency of their misbehavior known as occurrence counts whereas the latter list comprises the nodes declared as malicious on the basis of the occurrence counts in the ***suspicious list***. The analysis of REPORT packets obtained from the on-path nodes can result in one of the following conditions:

The N-bit flags in the REPORT packets of all the nodes have all zeros which indicates that first node in the list of on-path is part of the ***suspicious list***. If the N-bit flags does not contain all zeros, then the REPORT packets of each of the on-path nodes are analyzed sequentially. For each REPORT packet, a count is made of the number of packets for which the N-bit flag has a 1 but acknowledgement value from the downstream node is missing and the corresponding bit position is zero in the destination node's REPORT packet. If the number of such packets are greater than 20% of the total number of packets sent by the source node, then the ***suspicious list*** is checked to see if the node already exists in it. If yes, then the occurrence count of the node is incremented, otherwise the node is included in the ***suspicious list*** and its occurrence count is set to 1. After the composition of the ***suspicious list***, the ***malicious list*** is formed by looking out for those nodes in the ***suspicious list*** whose occurrence count has exceeded the threshold called as MAL_THRESH (set to 2) and including them into malicious list followed by their removal from the ***suspicious list***.

Another important functionality of **Reports Processing** phase is that, each pair of successive nodes (x, y) on the source to destination path such that y does not provide ACK packets for more than 20% of packets forwarded by x for more than one communication session has to be included in a list called as ***alert list pairs*** to inform the destination through the RREQ packet for the upcoming communication session.

This information of ***alert list pairs*** is needed by the destination so that, it can drop any RREQ packet if the route it followed has the pair (x, y) as successive nodes on the source to destination path. The justification for this dropping is that, during the process of sending the REPORT packet, each intermediate node x creates its own ***alert list*** so as to avoid any node y in the ***alert list*** as it's downstream neighbor by dropping any RREP received from such a node. Hence, to avoid the overhead with respect to time incurred in the route establishment because of the dropping of RREP packet followed by retransmission of RREQ packets, the destination node itself can take care not to establish any route involving the ***alert list pair*** (x, y) as the RREP will be dropped midway at node x if the route has y as it's downstream neighbor.

The reports analysis followed by the composition of ***suspicious list***, ***malicious list*** and ***alert list pairs***, is followed by the next successive communication session. This is done at the end of **Reports Processing** phase by having the source node check if it has already buffered N number of packets for transmission. Otherwise, it sets a timer which expires at a fixed time interval and then checks whether the input buffer has sufficient (N) number of packets for transmission failing which it resets the timer. After sufficient numbers of packets have accumulated in the input buffer, the source node

can initiate the next communication session. The next successive communication session may use the existing secure route or it may require a fresh secure route from the source to destination based upon the following conditions:

Condition 1: If the source node has the ***break sno*** field with its default value of −1 at the **Reports Request** phase and no nodes are included in the ***malicious list*** at the end of **Reports Processing** phase of the current communication session.

Condition 2: If the source node has the ***break sno*** field with its default value of −1 at the **Reports Request** phase and one or more nodes are included in the ***malicious list*** at the end of **Reports Processing** phase of the current communication session.

Condition 3: If the source node has the ***break sno*** field with a value greater than zero at the **Reports Request** phase indicating a link break during the **Data Transmission** phase.

Under Condition 1, the existing secure route from source to destination can be used for the next communication session. The allocation of memory by the intermediate nodes and the destination node for the next session is done through the CLEAR packets as they traverse the source to destination path during the **Reports Request** phase of the current communication session.

Under Condition 2 and Condition 3, the establishment of a new route is needed such that no nodes from the ***malicious list*** are included on the path. Also, the route should avoid any on-path successive nodes from the ***alert list pairs***. The information about the ***malicious list*** is disseminated through the **Blacklist Propagation** phase and ***alert list pairs*** information is embedded within the RREQ packet by the source node during the **Secure Route Establishment** phase. Figs. 1–5 show the pseudo code which depicts the procedures through which a node handles the reception of each of the different types of control packet based upon the proposed security mechanism.

```
// WHEN A NODE RECEIVES A RREQ PACKET FROM SOME NODE
// Following procedure is used

01 if current node's address not equals destination address field
02     if the source node of the packet is the current node
03         Drop the packet
04     endif
05     if the RREQ packet has a broadcast ID already seen
06         Drop the packet
07     endif
08     if the RREQ packet contains a blacklist field
09         Update the blacklist and the alert list pairs
10     endif
11     if the RREQ packet is received from a blacklisted node
12         Drop the packet
13     endif
14     Make an entry for the broadcast ID
15     Update the routing table for the reverse route
16     Further broadcast the RREQ packet.
17 endif
```

Figure 1: (Continued)

```
18 if the current node's address matches the destination address
19    if the RREQ packet has a bcast ID already seen
20       Drop the packet
21    endif
22    if the RREQ packet contains a blacklist field
23       Update the blacklist and the alert list pairs
24    endif
25    if the RREQ packet received from a blacklisted node
26       Drop the packet
27    endif
28    if the RREQ packet has a non-empty alert list pairs
29       Update the alert list pairs at the destination
30    endif
31    if the path has two successive nodes in alert list pairs
32       Drop the packet
33    endif
34    Make an entry for the broadcast ID
35    Update the routing table for the reverse route
36    Decrypt the random value r in the RREQ packet
37    Compute the ack values using the value r
38    Allocate the memory for the ack report current session
39    Delete the memory of ack reports of earlier sessions
40    Send the RREP packet with the encrypted ack values
41 endif
42 return
```

Figure 1: Reception of RREQ packet

```
// When a node receives a RREP packet from other node
01 if current node's address equals the destination address field
02 if the pid field <  last received RREP packet's pid
03    Discard the packet
04 endif
05 Make an entry for the forward route to the destination
06 if the report flag set to 1    // fresh route by RPTRQ pkt
07    Store the route and Retrieve the report
08 endif
09 if report flag = 0 and break sno >= 0
10    Send (N-break sno+1) pkts from backup buffer
11          Send (break sno - 1) pkts from the input buffer
12 endif
13 if report flag = 0 and break sno < 0
14       Send the N packets from the input buffer
15 endif
16 Store the N packets sent into the backup buffer.
17 Set a timer for a period required for sending N pkts
18 if RERR packet is received
19    Execute the module Reception of RERR packet
20 endif
```

Figure 2: (Continued)

```
21 if timer expired and break sno = -1
22         Unicast CLEAR packet along the S to D path
23         Set the timer for unicast
24         Wait for the REPORT packets
25     endif
26 if       timer       expired       and       break       sno       >=0
27     Broadcast the RPTRQ packet
28             Set the timer for broadcast
29         Start waiting for the REPORT packets
30 endif
31      if CLEAR timer expires and all REPORT pkts not rcvd
32     Broadcast the RPTRQ packet
33          Set the timer for broadcast
34                 Start waiting for REPORT packets
35      endif
36 if RPTRQ timer expires and all REPORT pkts not rcvd
37     Broadcast the RPTRQ packet
38     Set the timer for broadcast
39     Start waiting for the reception of REPORT packets
40     Goto step 37
41 else
42     Execute the module Process Reports
43 endif
44  endif
45  if the current node's address not equals the dest addr field
46 if the previous hop node is existing in the alert list
47     Drop the packet
48 endif
49 Decrypt the ack value provided by the destination
50 Allocate memory for the ack report of the current session
51 Delete memory of any earlier data transmission sessions
52 Make an entry for the forward route to the destination
53 Forward the RREP packet to the upstream node on the route
54  endif
55  return
```

Figure 2: Reception of a RREP packet

```
// When a node receives a RERR packet from other node
// Following procedure is used

01  if srcpkt field matches the address of the current node
02 Update the appropriate routing table entry
03 Update the break sno field
04 Free the packet
05  else
06      Update the appropriate routing table entry
07 Send the RERR packet to the nodes in the precursor list
08  endif
09  return

// When a node receives a REPORT packet from other node
// Following procedure is used
```

Figure 3: (Continued)

```
01  if the current node's address equals the destination address
02 Store the report in the apt index location of reports array
03 if the reports array is filled with all the REPORT packets
04      Execute the module Process Reports
05 endif
06  endif
07  if the current node's address not equals destination address
08 Forward the REPORT packet
09  endif
10  return

// When a node receives a CLEAR packet from other node
// Following procedure is used

01  if current node's address not equals destination addr field
02 Send the REPORT pkt along the reverse route
03 Forward the CLEAR packet towards the destination
04  endif
05  if the current node's address equals the destination addr field
06 Send the REPORT pkt along the reverse route
07  endif
08  return

// When a node receives CBR data packet from the upper layer
// For data transmission from the source to destination
01  Buffer the packets till number of packets is equal to N
02  Broadcast the RREQ packet with an encrypted random value
03  Set the timer for obtaining RREP
04  Wait for the RREP packet from the destination
04  if timer expires and RREP received
05      Execute the module Reception of RREP packet
06  else
07      Goto step 02
08  endif
09  return
```

Figure 3: Reception of a RERR, REPORT, CLEAR packets from some other node and reception of a data packet from the upper layer by the source node

```
// When the source node receives all the REPORT pkts
// Reports Processing   // Occ Count: Occurrence count
01  if the report rcvd from the dest has all 1s in the N-bit flag
02 Wait for the input queue to buffer sufficient no. of pkts (N)
03  else
04 for each successive pair of intermediate nodes (say (x, y) )
05    a ← num of pkts rcvd by node x and not rcvd by dest
06    b ← num of pkts rcvd by node x with missing acks from y
07    if b > 0.2 * a
08      if x and y exist in suspicious list
09          Occ count of x and y is incremented by 1
10          if Occ count of x  and y > MAL_THRESH
```

Figure 4: (Continued)

```
            Remove x and y from the suspicious list
            Include x and y in the malicious list
        endif
        else
            Include x and y in the suspicious list
            Set the occurrence  count of x and y to 1
        endif
    endif
endfor
for each successive pair of intermediate nodes (say (x, y) )
   c ← num of pkts rcvd by node x
   d ← num of pkts rcvd by node x with missing acks from y
    if d > 0.2 * c
     if same behaviour was exhibited in any past session
         Include x and y in alert list pairs
     endif
    endif
endfor
if the fresh route has no blacklisted nodes or alert list pairs
   Wait for the input queue to buffer sufficient no.of pkts (N)
   Initiate the next successive session
else
   Initiate new secure route discovery
   Wait for the input queue to buffer sufficient no.of pkts (N)
   Initiate the next successive session
endif
endif
return


// Following procedure is used before sending the ack report
 c ← num of pkts rcvd by current node
 d ← num of pkts rcvd by current node with missing acks
 if d > 0.2 * c
if the downstream node is included in suspicious list
    Include it in the alert list
else
    Include it in suspicious list
endif
endif
return
```

Figure 4: Modules for processing the reports by the source node and processing done by the intermediate nodes before sending the reports to the source node

```
// When a node receives a RPTRQ packet from other node
// Following procedure is used

if current node's addr not equals destination address
    if source node address in the pkt = current node's address
        Drop the packet
    endif
```

Figure 5: (Continued)

```
05    if the packet contains a broadcast ID already seen
06        Drop the packet
07    endif
08    Make an entry for the broadcast ID
09    Update the routing table for the reverse route
10    if the nodeList field contains the name of the current node
11        Execute the module PreReport Processing
12        Send a REPORT packet along reverse route
13        Decrement recvLength field in the RPTRQ packet
14    endif
15    if the recvLength field is greater than zero
16        Further broadcast the RPTRQ packet
17        Drop the packet
18    endif
19 endif
20 if the current node's address matches the destination address
21    if the RPTRQ packet contains a broadcast ID already seen
22        Drop the packet
23    endif
24    Make an entry for the broadcast ID
25    Update the routing table for the reverse route
26    if the nodeList field contains the name of the current node
27        Create an RREP packet with the ack report
28        Set the report flag set to 1
29        Send the RREP packet along the reverse route
27        Execute the module PreReport Processing
28    endif
29    Decrement the value of recvLength  field in the RPTRQ pkt
30    if the recvLength field is greater than zero
31        Further broadcast the RPTRQ packet
32    else
33        Drop the packet
34    endif
35 endif
36 return
```

Figure 5: Reception of a RPTRQ packet

4.3 Blacklist Propagation

The dissemination of information about the blacklisted nodes within the malicious list is done using a MALI packet comprising the identities of those nodes included into the malicious list. Whenever a node receives a MALI packet, it checks to see if it has not already received a MALI packet with the same broadcast id. If yes, the packet is discarded as a duplicate, otherwise it updates its blacklist to include all the nodes in the malicious list and further broadcasts it.

The significance of the blacklist is that, whenever a node receives an RREQ packet it first checks to see whether it has arrived from a blacklisted node. If yes, it is discarded, otherwise it is further propagated.

5 Optimizations to the Proposed Approach

After the analysis of the acknowledgement reports in the Reports Analysis phase, the next successive communication session requires a secure route from the source node to the destination node by avoiding the nodes in the malicious list and alert list pairs. The proposed approach initiates the route discovery phase afresh after the reception of reports from all the on-path nodes followed by the Reports Analysis phase. This may result in an increased control packet overhead especially if a link break occurs in the Data Transmission phase and the RPTRQ packet has to be broadcast throughout the network. In other words, the broadcast operation has to be performed in the form of RPTRQ packets for the reception of acknowledgement reports followed by two more broadcast operations. One broadcast operation is in the form of MALI packets for the Blacklist Propagation phase and yet another broadcast operation in the form of RREQ packets for the Secure Route Establishment phase for the next successive communication session resulting in a three-fold increase of the control packet overhead. A possible optimization for reducing the control packet overhead would be for the destination, to have the RREP packet with the acknowledgement report embedded within it sent to the source node as a response to RPTRQ packet so that it serves the two purposes of report reception and a fresh route establishment.

At the end of the Reports Analysis phase, the source node can check the fresh route for the presence of any nodes in the malicious list or any two successive nodes on the path from the alert list pairs. If the fresh route is free from all the nodes in the malicious list and the alert list pairs, then the source node can readily use that route without any broadcast of RREQ packets for the next communication session. Otherwise, it has to initiate the route discovery through the Secure Route Establishment phase.

Another possible optimization would be to include the malicious list of the current communication session in the RREQ packets of the Secure Route Establishment phase for the next successive communication session so as to eliminate the additional overhead incurred in the broadcast of MALI packets in the Blacklist Propagation phase.

6 Performance Analysis

The network simulator ns-2 [22] is used to analyze the performance of the proposed security scheme using a varying number of black hole nodes. The experimental parameters are enlisted in Tab. 1.

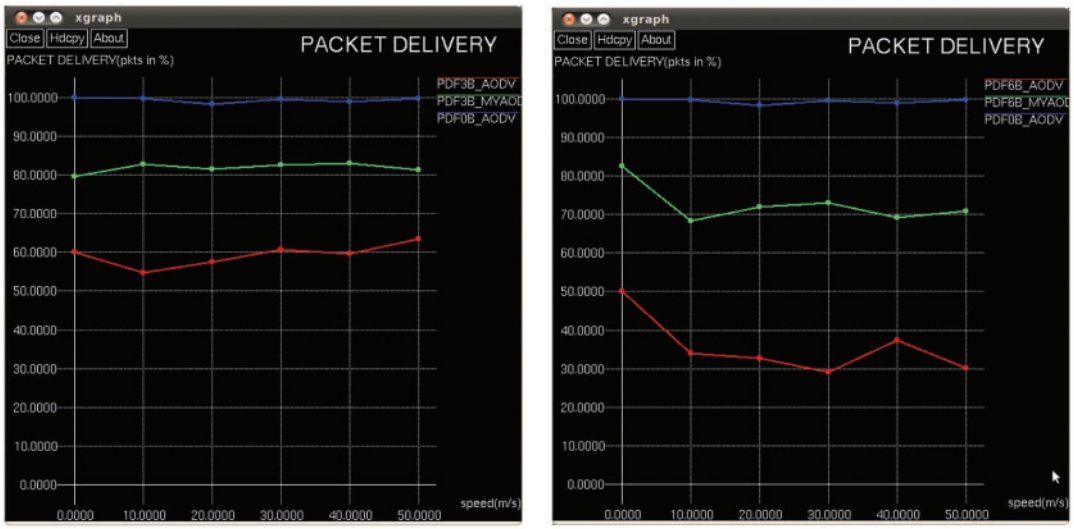
Table 1: Experimental parameters

Name of the parameter	Value of the parameter
Area of coverage	800 m × 800 m
Count of normal nodes	44 or 47
Count of malicious nodes	6 or 3
Range of transmission	150 m
Duration of simulation	1000 s
Model for mobility	Random way point
Type of traffic	User Datagram Protocol-Constant Bit Rate (UDP–CBR)
Size of packet	512 bytes
Node movement speed (m/s)	0, 10, 20, 30, 40, and 50
Pause duration	1 s

Two different forms of analysis were done wherein the first form had the normal nodes as well as malicious nodes move in a Random-way point mobility model, with 6 different types of speed, 0, 10, 20, 30, 40, and 50 m/s with a pause time of 1 s. The second form of analysis was done by considering the malicious nodes as stationary and the Random-way point mobility model was applied to all the remaining innocent nodes with 6 different types of speed, 0, 10, 20, 30, 40, and 50 m/s with a pause time of 1 s. Each data value in the figures refers to average value obtained through 15 experimental runs. In each form of analysis, we consider two cases, one with 3 malicious nodes and another with 6 malicious nodes. The packet delivery fraction with 3 and 6 black hole nodes respectively is depicted in the Figs. 6a and 6b respectively when the black hole nodes are moving as other normal nodes. In Fig. 6a, it can be observed that the mean packet delivery fraction for all speeds is 99.3% when there is no black hole node but it sharply falls down to 61.47% in the presence of 3 black hole nodes. The usage of the proposed security mechanism results in an improved mean packet delivery fraction of 81.13%.

Figure 6: (a) Packet delivery fraction for 3 moving black hole nodes **(b)** Packet delivery fraction for 6 moving black hole nodes

Fig. 6b, shows that the mean packet delivery fraction for all speeds is 99.3% when there is no black hole node but in the presence of 6 black hole nodes, it sharply falls down to 38.09%. The Packet Delivery Fraction for the second form of analysis involving stationary black hole nodes is depicted in Figs. 7a and 7b. In Fig. 7a, it can be observed that the mean packet delivery fraction for all speeds is 99.3% when there is no black hole node but it sharply falls down to 59.36% in the presence of 3 stationary black hole nodes. The proposed security mechanism generates an improved mean packet delivery fraction of 81.76%. Fig. 7b, shows that the mean packet delivery fraction for all speeds is 99.3% when there is no black hole node but in the presence of 6 black hole nodes, it sharply falls down to 35.54%. An improved mean packet delivery fraction of 72.59% can be achieved through the proposed security mechanism.



(a) (b)

Figure 7: (a) Packet delivery fraction for 3 stationary black hole nodes **(b)** Packet delivery fraction for 6 stationary black hole nodes

The true positive rate and the false positive rate as an outcome of experiments conducted considering moving black hole nodes and stationary black hole nodes is depicted in Figs. 8 and 9 respectively. As can be seen in Figs. 8a and 8b, the mean True Positive rates are 66.6% and 76.7% and the mean false positive rates are 3.2% and 6.3% respectively for 3 moving black holes and 6 moving blackholes.

(a) (b)

Figure 8: (a) True/False positive rate for 3 moving black hole nodes **(b)** True/False positive rate for 6 moving black hole nodes

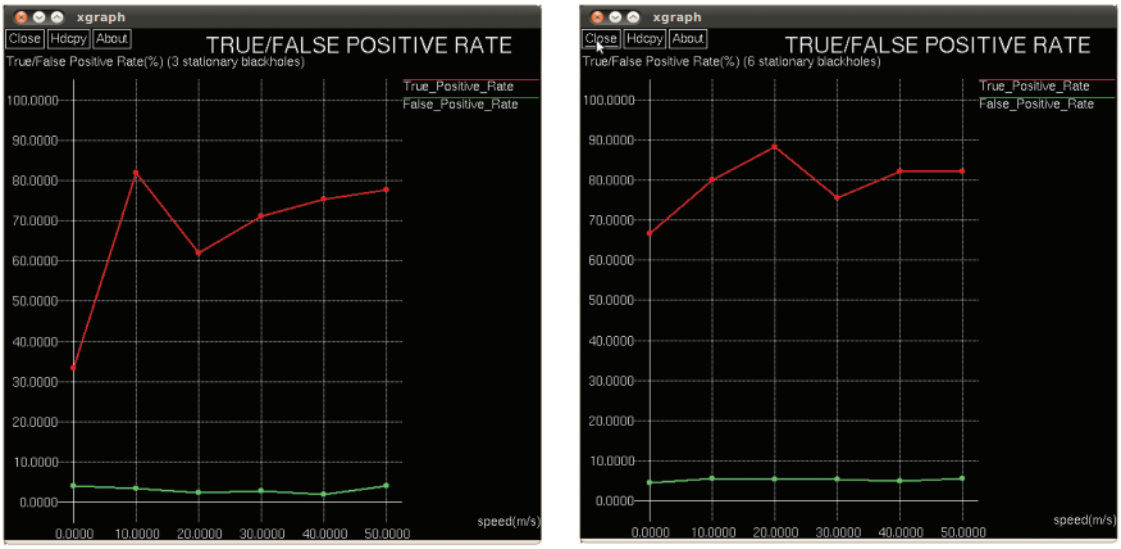


Figure 9: (a) True/False positive rate for 3 stationary black hole nodes **(b)** True/False positive rate for 6 stationary black hole nodes

As can be seen in Figs. 9a and 9b, the mean True Positive rates are 66.8% and 79%. The mean false positive rates are 3.2% and 5.2% respectively for 3 stationary black holes and 6 stationary black holes.

7 Conclusion and Future Work

In the presence of multiple number of black hole nodes, a node which is not yet detected as a black hole can reduce the packet delivery fraction of the sessions until it is detected.

The approach given in [10] works during the route discovery and route establishment phases by considering the destination sequence number (DSN) and the IP address of the node sending the RREP packet. It also takes care of reducing the false positives by having an additional field in the routing table. A comparison of the proposed approach with the one in [10], the following observations can be drawn:

The former approach focuses upon the malicious node detection during the route establishment phase whereas the current research involves an approach based upon historical behavioral analytics of packet forwarding behavior during data transmission phase after the route establishment. This approach has significance in a situation involving intelligent malicious nodes adopting various strategies to evade detection during the route establishment and exhibit the malicious behavior after forming the route.

The future work aims to design a mechanism which can be incorporated in the proposed security mechanism to determine a reputation rating for each intermediate node on the path by each source node which can be used assign different relative weights to a pair of successive nodes (a, b) on the source to destination path exhibiting the suspicious behavior in a session so as to ensure the innocent node is not falsely detected as malicious and the malicious node is truly detected as a black hole node.

Acknowledgement: The author would like to thank Deanship of Scientific Research at Majmaah University for supporting this work under Project Number 1439-59.

Funding Statement: The author received funding for this research from Deanship of Scientific Research at Majmaah University for supporting this work under Project Number 1439-59.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Y. Su, "Prevention of selective black hole attacks on mobile ad hoc networks through intrusion detection systems," *Computer Communications*, vol. 34, no. 1, pp. 107–117, Jan 2011.

[2] H. Choi, W. Enck, J. Shin, P. McDaniel, and T. F. La Porta, "Secure reporting of traffic forwarding activity in mobile ad hoc networks," in *Proc. 2nd Annual Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services, MobiQuitous*, San Diego, CA, USA, pp. 12–21, 2005.

[3] T. A. Amornkul, "On detection mechanisms and their performance for packet dropping attack in ad hoc networks," Ph. D Thesis, University of Pittsburgh, 2008.

[4] S. S. Zhanga, S. Wanga, H. Xiaa, and X. Chenga, "A review on propagation of secure data, prevention of attacks and routing in mobile ad-hoc networksmobile ad hoc networks," *Procedia Computer Science*, vol. 147, pp. 473–479, 2019.

[5] N. Asai, S. Goka, and H. Shigeno, "A Reputation-based model for trust evaluation in social cyber-physical systems," *IEEE International Workshop on Pervasive Flow of Things*, vol. 7, no. 2, pp. 517–522, 2019.

[6] H. Xia, B. Li, S. Zhang, S. Wang and X. Cheng, "A novel recommendation-based trust inference model for MANETs," in *Lecture Notes in Computer Science*, Springer, Cham: Springer, vol. 10874, pp. 893–906, 2018.

[7] W. Alnumay, U. Ghosh, and P. Chatterjee, "A Trust-based predictive model for mobile ad hoc network in internet of things," *Sensors*, vol. 19, no. 6, 1467, pp. 2019.

[8] G. M. Borkar and A. R. Mahajan, "A Trust-based predictive model for mobile ad hoc network," *Int. J. Communication Networks and Distributed Systems*, vol. 24, no. 1, pp. 23–57, 2020.

[9] M. Rmayti, R. Khatoun, Y. Begriche, L. Khoukhi, and D. Gaiti, "A stochastic approach for packet dropping attacks detection in mobile Ad hoc networks," *Computer Networks*, vol. 121, pp. 53–64, 2017.

[10] S. Sivanesh, and V. R. S. Dhulipala, "Accurate and cognitive intrusion detection system (ACIDS): A novel black hole detection mechanism in mobile ad hoc networks," *Mobile Network Applications*, vol. 26, pp. 1696–1704, 2021.

[11] T. Poongodi, S. M. Khan, R. Patan, A. H. Gandomi, and B. Balusamy, "Robust defense scheme against selective drop attack in wireless ad hoc networks," *IEEE Access*, vol. 7, pp. 18409–18419, 2019.

[12] A. F. Subahi, Y. Alotaibi, O. I. Khalaf, and F. Ajesh, "Packet drop battling mechanism for energy aware detection in wireless networks," *Computers, Materials & Continua*, vol. 66, no. 2, pp. 2077–20862, 2021.

[13] V. H. Kshirsagar, A. M. Kanthe and D. Simunic, "Trust based detection and elimination of packet drop attack in the mobile ad-hoc networks," *Wireless Personal Communications*, vol. 100, no. 2, pp. 311–320, 2018.

[14] A. B. Usman, W. Liu, Q. Bai and A. Narayanan, "Trust of the same: Rethinking trust and reputation management from a structural homophily perspective," *International Journal of Information Security and Privacy (IJISP)*, vol. 9, no. 2, pp. 13–30, 2015.

[15] Bisen D. and Sharma S., "Fuzzy based detection of malicious activity for security assessment of MANET," *Natl. Acad. Sci. Lett*, vol. 41, pp. 23–28, 2018.

[16] E. Elmahdi, S. Yoo, and K. Sharshembiev, "Secure and reliable data forwarding using homomorphic encryption against blackhole attacks in mobile ad hoc networks," *Journal of Information Security and Applications*, vol. 51, pp. 102425, 2020.

[17] K. Vanitha and A. M. J. Z. Rahaman, "Preventing malicious packet dropping nodes in MANET using IFHM based SAODV routing protocol," *Cluster Computing*, vol. 22, pp. 13453–13461, 2019.
[18] S. Mukherjee, M. Chattopadhyay, S. Chattopadhyay, and P. Kar, "EAER-AODV: Enhanced trust model based on average encounter rate for secure routing in MANET," *Advanced Computing and Systems for Security*, vol. 667, pp. 135–151, 2018.
[19] S. R. Halhalli, S. R. Sugave, and B. N. Jagdale, "Optimisation driven-based secure routing in MANET using atom whale optimization algorithm," *International Journal of Communication Networks and Distributed Systems*, vol. 27, no. 1, pp. 77–99, 2021.
[20] V. V. Kumar, and S. Ramamoorthy, "Secure adhoc on-demand multipath distance vector routing in MANET," in *Lecture Notes in Networks and Systems*, Singapore: Springer, vol. 24, pp. 49–63, 2018.
[21] C. E. Perkins, E. B. Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," IETF Internet Draft, MANET working group, 2004.
[22] The network simulator-ns-2, 1989. http://www.isi.edu/nsnam/ns/.