Trust based Routing using Dominating Set Approach (TRDSA) in Wireless Ad-Hoc Networks

Deepika Kukreja
Research Scholar, USIT
G.G.S.I.P.U., Delhi, India

Umang
Research Scholar, USIT
G.G.S.I.P.U., Delhi, India

B. V. R. Reddy
Professor, USIT
G.G.S.I.P.U., Delhi, India

ABSTRACT
Secure routing protocols for Mobile Ad hoc Networks (MANETs) have been categorized based on the model used for enforcing security, methodology and information they use to make routing decisions. Some protocols are designed from scratch so as to incorporate security solutions and some are designed to provide security mechanisms into the existing routing protocols like DSDV[1], OLSR[2], AODV[3], DSR[4] etc. Several protocols for secure routing in ad-hoc networks have been proposed. But due to their limitations, there is a need to make them robust and more secure so that they can go well with the demanding requirements of ad hoc networks. We propose and design a new protocol - Trust based Routing using Dominating Set Approach (TRDSA) which overcomes the shortcomings of existing protocols.

General Terms
Design, Security.

Keywords
Routing protocol, Trust, Mobile Ad Hoc Networks, Secure routing protocols.

1. INTRODUCTION
A mobile ad-hoc network is a self-configuring network of mobile hosts connected by wireless links which together form an arbitrary topology. A secure ad hoc network has to meet several security requirements. Several protocols have been designed for making ad hoc routing secure and robust. But due to lack of centralized control, dynamic network topology, high power consumption, low bandwidth, high error rates and multihop communications, the provision of making routing secure in mobile ad hoc networks is much more challenging than the routing security in infrastructure based networks. Most of the existing work [5-12] in the area of secure routing protocols in an ad hoc network is based on key management, heavy encryption techniques or on continuous promiscuous monitoring of the neighbors. These approaches for making ad hoc routing secure are expensive due to which they do not fit well for MANET. In this paper, we propose and design a new protocol - Trust based Routing using Dominating Set Approach (TRDSA) which overcomes the shortcomings of existing protocols.

The rest of the paper is structured as follows: Section 2 presents an overview of secure routing protocols in ad hoc networks. A novel Trust based Routing protocol using Dominating Set Approach (TRDSA) is proposed and presented in section 3. In Section 4, the proposed routing protocol TRDSA is illustrated. Section 5 concludes the paper.

2. OVERVIEW OF SECURE ROUTING PROTOCOLS IN AD HOC NETWORKS
There are several secure routing protocols in the literature that were designed to cope with the limitations and requirements of ad hoc networks. Some of which are based on trust and some are not based on trust.

2.1 Non-trust based secure routing protocols
Marti et al. designed Watchdog and Pathrater mechanism [8] to optimize the packet forwarding method in the Dynamic Source Routing (DSR) protocol [4]. It consists of two components: Watchdog and Pathrater. The Watchdog detects selfish nodes that do not forward packets and the Pathrater helps routing protocols to avoid these nodes. It assigns ratings to the nodes, based upon the feedback it receives from the Watchdog. These ratings are then used to select routes having nodes with the highest forwarding rate. Watchdog's weaknesses are that it might not detect a misbehaving node in the presence of: Ambiguous collisions, Receiver collisions, Limited transmission power, False misbehavior and Partial dropping.

CONFIDANT (Cooperation of Nodes, Fairness In Dynamic Ad hoc NeTworks) [9] adds a trust manager and a reputation system to the Watchdog and Pathrater mechanism [8]. The trust manager evaluates the events reported by the Watchdog and in order to warn other nodes in the network regarding malicious nodes (for not forwarding), it sends alarm. The reputation system maintains a black-list of nodes at each node and shares this list with the nodes in its friends-list. The CONFIDANT protocol is based on a punishment scheme, by not forwarding packets of nodes whose trust level drops below a certain threshold.

Dahill et al. proposed ARAN (Authenticated Routing for Ad-hoc Networks) [10] that detects and protects against misbehaviors of malicious nodes in an ad-hoc network. ARAN is based on asymmetric cryptography, make uses of digital certificates and all nodes are supposed to keep fresh certificates with a trusted server and should know the server's public key. ARAN requires the use of a trusted certificate server in the network which is against the nature of MANETs.

Y.Hu. et al. proposed SEAD (Secure Efficient Ad hoc Distance vector) [11], based on Destination Sequenced Distance Vector (DSDV) [1] protocol. It uses one way hash function and authentication to differentiate between updates received from malicious and non-malicious nodes. It overcomes the DoS and resource consumption attacks but fails when the attacker uses the same metric and sequence number as used by recent update message. In SEAD nodes have hash chain which has a finite size and must be generated again when all their elements have been used.

Y.Hu et al. [12] proposed ARIADNE, an on-demand secure routing protocol based on the Dynamic Source Routing (DSR)

that protects against node compromise. It is based upon symmetric cryptography and the distribution of shared secret keys between source and the destination. For node authentication ARIADNE prefers using the TESLA [13] broadcast authentication scheme with delayed key disclosure. TESLA requires clock synchronization between communicating nodes and this requirement is unrealistic in MANETs.

2.2 Trust based secure routing protocols

The existing work in the area of trust based secure routing in ad hoc networks as proposed in [14-27] require each network node to work in promiscuous mode. Working in the promiscuous mode requires nodes to have high energy capacity as they need to overhear all the transmissions and promiscuous listening also increases the network overhead. In this paper, we propose and design a new trust based secure routing protocol which overcomes the shortcomings of existing protocols in this area.

3. TRUST BASED ROUTING USING DOMINATING SET APPROACH (TRDSA)

The proposed protocols in [18-20, 23, 27] fail to work when malicious nodes collude together in order to harm the network. Existing protocols proposed in [14-27] require each network node to work in promiscuous mode. Working in the promiscuous mode requires nodes to have high energy capacity as they need to overhear all the transmissions and promiscuous listening also increases the network overhead. Protocols [19, 24] detect malicious nodes inducing only one or two types of attacks. Protocols proposed in [14, 15, 18, 22] increase delay in route discovery. In [14, 15], proposed protocols result in the most trustworthy path but do not discover the shortest path. Some protocols require each node to have high memory capacity as they store large tables used for storing security information. So there is a need to design a secure routing protocol which overcomes the above mentioned limitations of the existing trust based routing protocols.

TRDSA presents a method for selection of a trustworthy and shortest path between source and destination that is free from malicious nodes. TRDSA is able to detect the malicious nodes inducing attacks like: grayhole, malicious topology change behavior, dropping data packets, dropping control packets, modifying the packet and malicious flooding. The solution also works for attacks induced by colluding malicious nodes. In TRDSA, unlike most of the existing protocols in this area, it requires only few nodes to operate in promiscuous mode which results in the reduction of network overhead as compared to the protocols which requires all the network nodes to work in promiscuous mode. Nodes working in the promiscuous mode persistently require nodes to have high energy capacity as they need to overhear all the transmissions. We have referred [28] to select a set of n nodes called Dominating set such that all the nodes in the network are either in the Dominating set or neighbors of the nodes in the Dominating set. From the Dominating set l most trusted nodes (initially trust represents less mobile nodes) that have remaining energy higher than threshold energy required for working in the promiscuous mode are selected. These selected nodes are called Leader nodes (L1, L2, L3……..Ll). The remaining n-l nodes are called Assistant nodes (A1, A2….) which operate in promiscuous mode only when the need arises, this saves nodes' energy and the rest other nodes in the network which are not in the Dominating set are called Regular nodes.

3.1 Terminology

m_threshold: It is the minimum value of the trust level below which a node is considered as malicious.

s_threshold: It is the trust level below which a node is considered as suspicious and m_threshold < s_threshold.

L_energy: It is the minimum remaining energy required for a node to persistently operate in promiscuous mode and to act as a Leader.

L_trust: It is the minimum required trust level for a Leader node.

A_energy: It is the minimum remaining energy required for a node to work as an Assistant node.

A_trust: It is the minimum required trust level for an Assistant node.

3.2 Trust Computation

T. Ghosh et al [15] while computing trust of a node takes the mobility of a node into account and makes the confidence level of a node as zero if the mean of time difference of that node leaving the network, μ is below a threshold value. Otherwise it is independent of mobility. Here instead of taking malicious index of a node x, M(x) value as either zero or one, we consider that M(x) lies in the range [0, 1]. 0 means the highly mobile node inducing malicious topology change behavior and 1 means a low mobility node as required for Leader nodes and 0.5 means a node with moderate mobility.

The malicious index of node A, M(A) is computed as:

$M(A) = (1-e^{-\mu/\lambda})$.. …………………………………...(1)

Where μ is the mean of the time difference of a node leaving the network computed as in [15]. λ is the factor by which μ is related to the malicious index of node A, M(A). It depends upon the nodes' mobility. In our work, we consider λ =10 for μ = 0 to 200. The malicious index of node A, M(A) will be near to 0 for a mobile node changing network topology frequently and it is near to 1 if the mobile node is stable.

Leader nodes compute the trust of their neighboring nodes as:

TL(A) = Forward_Behavior x M(A) ………………...(2)

Where

Forward_Behavior = (HF(A) x Pkt_size(HF(A))) / (SF(A) x Packet size(SF(A)))

SF(A): The total number of packets sent by all nodes to node A for forwarding. It is given by:

SF(A) = n∑i=1SFi(A), SFi(A) is the total number of packets sent by node i to node A for forwarding and n is total number of neighbors of node A.

HF(A): The total number of packets that have been forwarded by node A.

Pkt_size(HF(A)): Total size of the packets forwarded by node A.

Pkt_size(SF(A)): Total size of the packets sent by all nodes to A for forwarding.

Different Leaders may compute different trust values for the same node (if node is a neighbor of more than one Leader) according to their experience and interactions with the node; in

this case the combined trust is computed based on the number of interactions as follows:

$$T(A) = p\sum i=1(NLi(A)*TLi(A)) / p\sum i=1NLi(A) \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Where

T(A) is the combined trust of node A, p is the number of neighbor Leaders of node A, NLi(A) is the number of interactions of Leader node Li with node A and TLi(A) is the individual trust of node A computed by Leader node Li using equation 2.

All neighbor Leaders of node A store the value of combined trust rather than individual trust about node A in their trust table.

3.3 Algorithm for the selection of Leader, Assistant and Regular nodes

The following algorithm is executed periodically for the selection of Leader, Assistant and Regular nodes:

```
If (node i belongs to the dominating set)
{
        If (energy (i) >= L_energy and trust (i) >= L_trust)
{
                        Set status (i) = Leader
                        Promiscuous (i) = ON
                    }
Else if (energy (i) >= A_energy and trust (i) >=      A_trust)
                Set status (i) = Assistant
Else
                        Set status (i) = Regular
          }
         Else
           Set status (i) = Regular
```

It is assumed that initially all the nodes in the network have similar and enough remaining energy to act as a Leader or an Assistant node. As working in promiscuous mode is expensive, only few nodes called the Leader nodes work in promiscuous mode and they overhear the transmissions in its range. Each Leader and Assistant node maintains a Trust_Table having three fields: NodeID, status and Trust_level. NodeID is the unique ID of a node, Status of a node has values like Leader, Assistant or a Regular node, Trust_level is the trust value of a node computed by the Leader or received from other Leader nodes. During route discovery Leader nodes append the trust level of all the nodes that RREQ packet has traversed to the RREQ packet. Leaders exchange the Trust_Table periodically with the Assistant nodes and other Leaders in their neighborhood.

In the proposed protocol TRDSA, each node is under observation: A single Leader node is observed by Assistant nodes when required and by other neighboring Leader nodes. Assistant nodes are monitored by Leader nodes. All Regular nodes are observed by Assistant nodes (if in promiscuous mode) or Leader nodes or both. Leaders exchange the Trust_Table periodically with the Assistant nodes and other Leaders in their neighborhood. The Leader nodes consume more energy as they are persistently operating in the promiscuous mode. Thus the remaining energy of the Leader nodes keeps on decreasing with time. The Asistant nodes operate on and off in promiscuous mode. So their energy usage is less as compared to energy usage of Leader nodes.

As Leader nodes are working in the promiscuous listening mode they may learn multiple routes to any destination and store them in their cache. This avoids the additional route discoveries in case of route breaks.

3.4 Route Discovery

Modified version of DSR protocol is used for discovering multiple partial disjoint paths [29] during route discovery. Discovery of multiple paths reduces end-end-delay and it quickly recovers in case the route breaks. Source node broadcasts a Route Request message (RREQ) containing Source address, Request ID, Source Sequence number, Destination address, Route record, Trust of nodes in the route record and Time to Live (TTL) counter.

In TRDSA, the receiving node discards the RREQ packet if any of the following five conditions is satisfied:

1. If Time to Live counter of the RREQ reaches zero.
2. If the ID of the receiving node is already present in the route record of RREQ.
3. If trust of any node in the route record of RREQ is below the suspicious threshold.
4. If the received RREQ is duplicate and the same route record already exists in the cache.
5. If the received RREQ is duplicate and the number of hops of the route record is greater than the previously entertained RREQ.

If the node has received the request for the first time, it computes the hop count of the route record, stores the hop count in cache as HopCntMin, and adds its own address to the route record in RREQ and then rebroadcast the RREQ. When a node receives duplicate RREQ packet, the node rebroadcast the RREQ after adding its own address, if the hop count of the route record in received RREQ is less than or equal to the stored HopCntMin. The value of the stored HopCntMin is also changed to the newly computed Hop count. Otherwise, the node drops the RREQ packet. Thus the method discovers the shortest paths and also avoids the RREQ packets storm. If the sender of the RREQ is the source then the RREQ packet is processed without checking the trust of the sender otherwise before processing the RREQ packet the trust of the sender is first verified, if the trust of the sender is below the suspicious threshold then the packet is dropped. If the receiving node is a Leader, then it checks the trust of the nodes present in route record of RREQ in its trust table, if trust of any node is less than the suspicious threshold then the RREQ packet is discarded. If the sender of RREQ is also a Leader node and the trust of the sender Leader node is less than the suspicious threshold than the receiver Leader node invokes the procedure for malicious detection of the sender Leader node. When a node receives RREQ packet from a Leader node, it updates the trust of the nodes in its trust table (if Leader or Assistant) and cache using trust of the nodes in route record.

In TRDSA, only Leader and Assistant nodes are allowed to append trust level of other nodes to route discovery packets. As less trusted nodes are not allowed to provide trust level of other nodes and also they themselves cannot declare a node as malicious. So the method avoids the attack induced by malicious colluding nodes. Also, only Leader intermediate nodes are allowed to send RREP if they have path to the destination. This further prevents the case where malicious node sends RREP in order to induce attack in the network. When the destination node receives the RREQ, it sends the RREP through the reverse path. RREP contains the trust of the nodes in reverse route along with the other fields.

The proposed method TRDSA, reduces the number of route discovery control packets by dropping the packets immediately

on detection of malicious nodes during route discovery. Thus, the source route cache stores only safe routes free of malicious

nodes and selects the most trustworthy and shortest path from the discovered routes.

3.4.1 Procedure for malicious detection of Leader

In TRDSA, the Leaders are selected in such a way that each Leader is being monitored by one or more other Leaders, when a Leader (accuser Leader) detects its neighboring Leader (accused Leader) as suspicious, instead of sending alarm message to other nodes it first sends report about malevolent behavior of the accused Leader to the Leader (monitor Leader) which is the neighbor of both accused Leader as well as accuser Leader. The monitor checks the trust levels of both the accuser and the accused Leaders in its trust table. The Leader having lesser value of trust is declared as malicious Leader and monitor Leader sends an alarm message about this malicious Leader to all the other nodes in the network. Detected malicious Leader is isolated from the network. If no such Leader which is the neighbor of both accused as well as accuser Leaders exists than the Assistant node (Monitor Assistant node) which is the neighbor of both accused and accuser Leaders is selected for monitoring. Monitor Assistant node starts working in the promiscuous listening mode and as it already has trust information about both these Leaders in its trust table. Based on the trust information as well as observed behavior it decides the trustworthy and malicious Leader and sends this decision to its own neighboring Leader which further sends an alarm message about this malicious Leader to all the other nodes in the network. Detected malicious Leader is then isolated from the network and monitor node goes back in the non promiscuous mode.

3.4.2 Path Trust Computation

All the discovered paths stored in source cache are free from malicious nodes and the source node computes the Path Trust, PTi. The paths are assigned weights such that, the path having lesser number of hops as compared to other paths is given more weight. In equation (4), a path will have more Path Trust, if the minimal trust level of the nodes in the path is more than that in other discovered routes and also the path has lesser number of hops as compared to other discovered routes.

$T_i = T_{avg} – (T_{avg} – T_{mini}) / (T_{mini} – m\_threshold)$
$PT_i = T_i \times {}^{q}\sum_{i=1} q_i / q_i$ ……………………………...…(4)

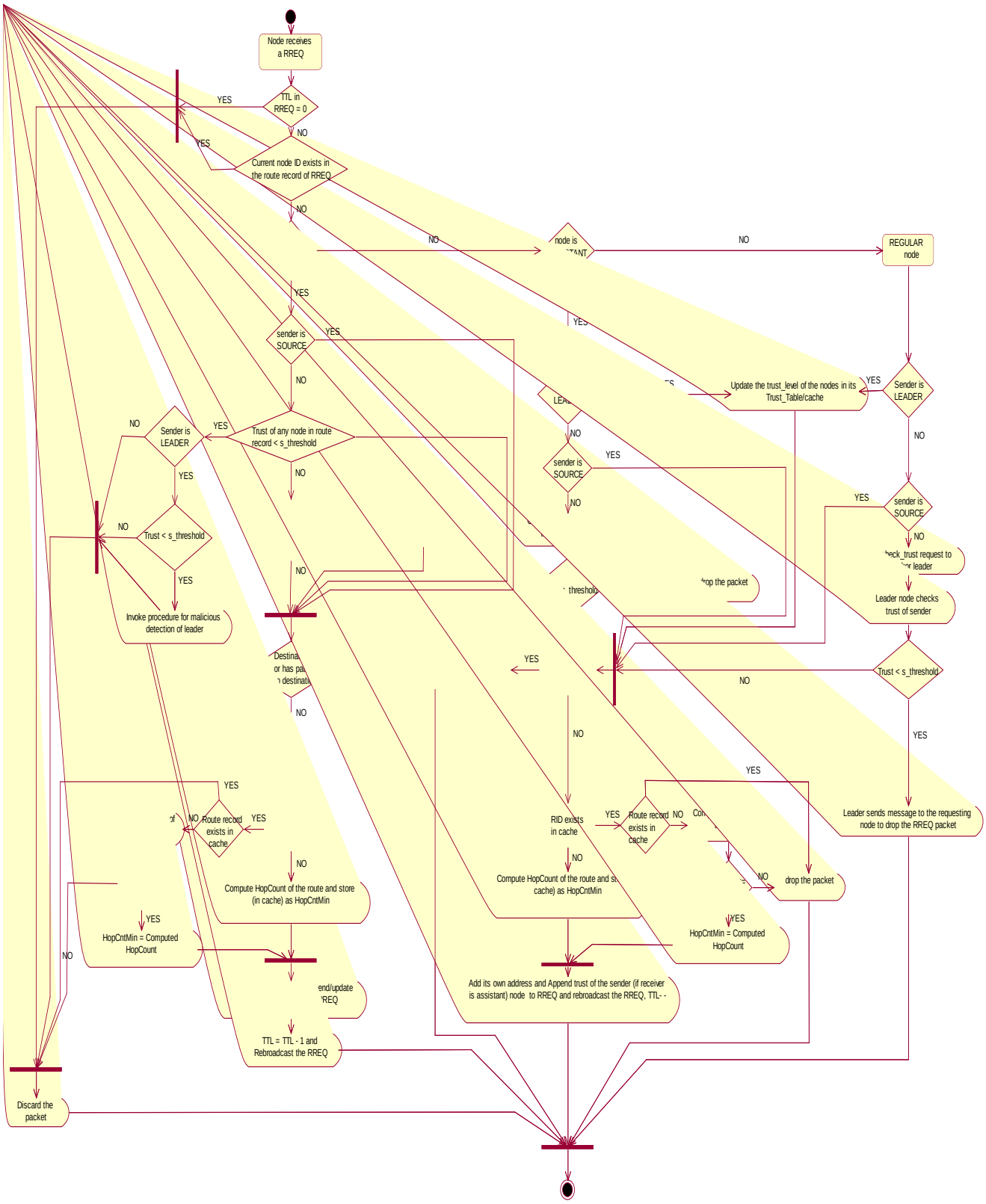

Fig. 1: Process flow for a node when it receives Route Request (RREQ) Packet

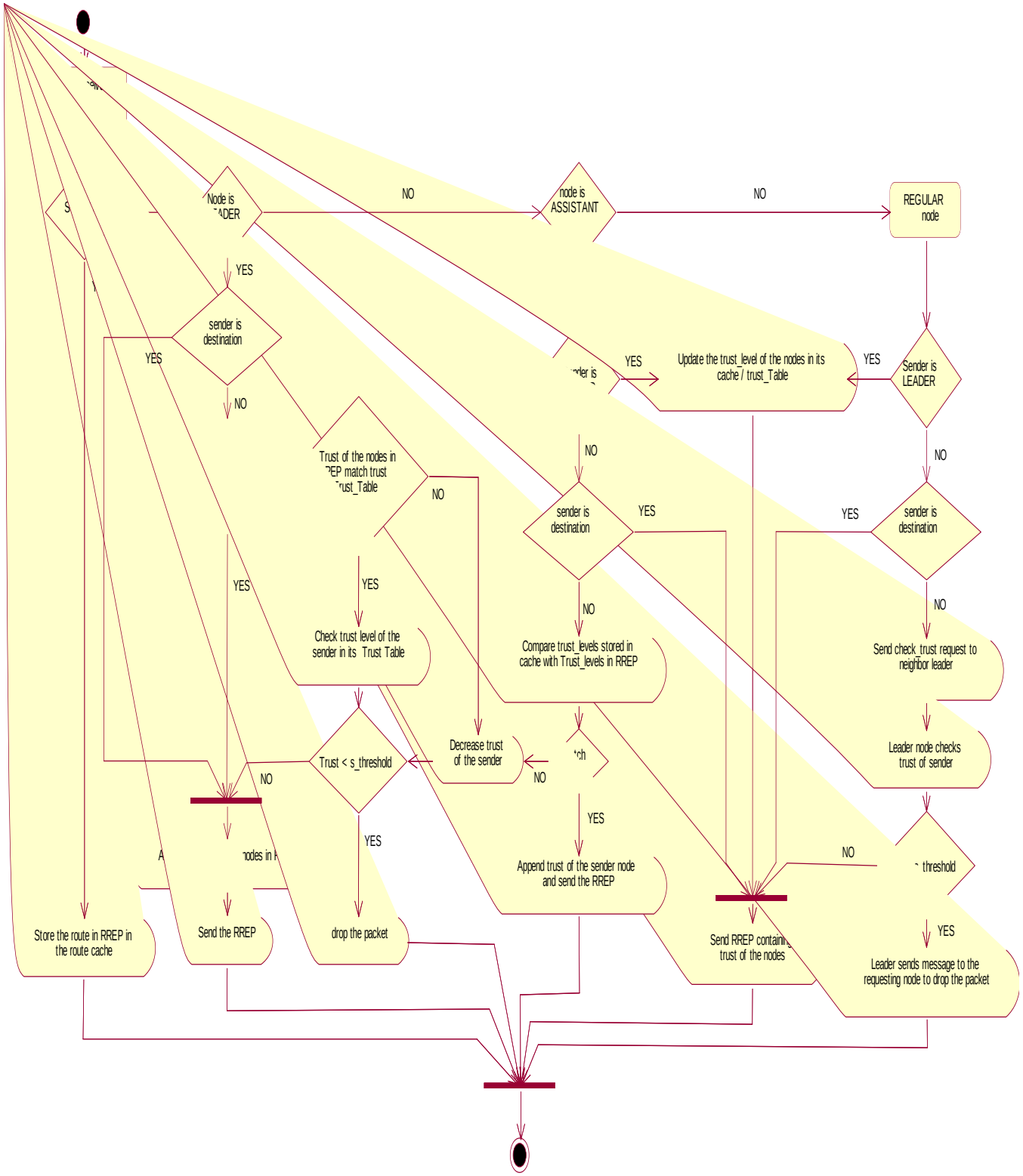

Fig 2: Process flow for a node when it receives Route Reply (RREP) Packet.

Where T_i is the trust score, T_{avg} is the average of trust levels of nodes in the path i and T_{mini} is the minimal trust level of a node in the path i, q is the total number of discovered paths and q_i is the total number of nodes in the path i.

3.4.3 Process Flow

The process flow diagram for route discovery is shown in fig. 1 and the process flow diagram when a node receives Route Reply packet shown in fig. 2.

3.5 Algorithm for Routing Data Packets

```
if (source node has data to send to the desired destination)
{
if (routing cache is empty)
    initiate route discovery
else
{
 scan the routing cache for the desired destination
if (alternate routes found)
Compute Path Trust for all routes and send the data  packet through the route having maximum PT
// Path trust is more for the most trustworthy as well as shortest  routes
else
                Initiate route discovery
}
}
```

During the transmission of data, Leader nodes in the selected path monitor their neighbors and decrease or increase their trust levels accordingly. Leader nodes as working in promiscuous mode also learn about path breaks and drop the data packet and send RERR message if path is broken. The nodes receiving the RERR message remove this link from their cache.

4. ILLUSTRATION

In fig 3, nine nodes (red and blue coloured nodes) are selected using [28] that cover the whole network. Out of these nine nodes, five nodes having high remaining energy and less mobility are selected as Leader nodes and rest four nodes act as Assistant nodes by executing algorithm 1.

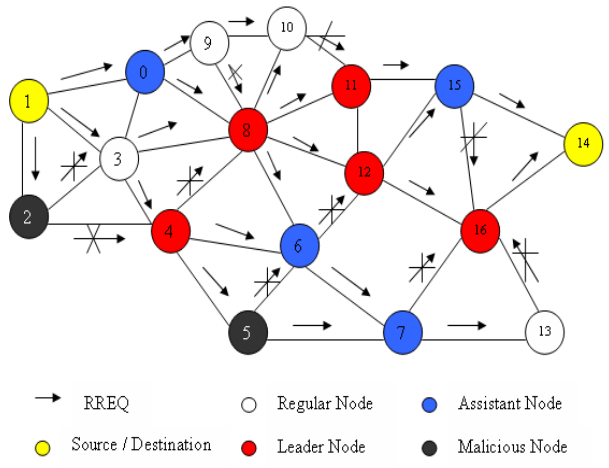

Fig 3: Network of seventeen nodes

Nodes numbered as 4, 8, 11, 12 and 16 are the Leader nodes that persistently work in promiscuous listening mode, run watchdog and compute the trust of their neighboring nodes. Nodes 0, 6, 7 and 15 are the assistant nodes which run watchdog only when it is required. Leaders exchange their trust tables with each other and with other assistant nodes periodically. We assume that nodes 2 and 5 are malicious. Suppose node 1 wants to send data to node 14. Before sending the data, source node (node 1) checks its cache to search a route to the desired destination, as no route found, it initiates Route discovery by broadcasting Route Request Packet (RREQ) to its neighboring nodes 0, 2 and 3 having route record containing only node 1. On receiving RREQ, nodes 0, 2 and 3 check the source ID in RREQ and as sender of the RREQ packet is the source node, nodes 0, 2 and 3, compute and store hop count in their cache, add their own addresses to the RREQ and rebroadcast the RREQ packet containing route records as 1-0, 1-2 and 1-3 respectively. Node 3 drops the duplicate RREQ received from node 2 as Hopcount of the reverse route 3-2-1 is greater than the stored HopCntMin. As node 4 is a Leader node running watchdog, it checks the trust level of the sending node 2, it finds that trust of node 2 is below the m_threshold, so, node 4 sends an alarm message to all the other nodes about the malicious node 2. On receiving this alarm message, node 3 also update the trust level for node 2 in its cache and node 3 and node 4 drop all the data and control packets coming from node 2 and node 2 is isolated from the network services. Node 4 receives RREQ from node 3, it checks the trust of node 3, as it is greater than s_threshold and it does not has path to the destination node 14, it stores the reverse route with Request ID, compute HopCountMin, Adds its own address and appends the trust of node 3 in RREQ and rebroadcasts the RREQ having route record as 1-3-4. Node 9 receives the RREQ from node 0, it sends request to Leader node 8 for checking trust of node 0, as trust of node 0 is greater than s_threshold, node 9 computes the hop count of the path as HopCountMin, stores this value in its cache and adds its own address and rebroadcast the RREQ having route record as 1-0-9. Let node 8 receives first RREQ from node 0; it computes the hop count of the path 8-0-1 as HopCountMin = 3 and it stores this value in its cache and adds its own address and trust of node 0 and then broadcasts this RREQ. Node 8 receives one another RREQ from node 3; it computes the hop count of the path 8-3-1 and compares it with the stored HopCountMin in its cache, as the computed hop count is equal to the stored HopCountMin, it adds its own address and trust of node 3 and then it broadcasts this duplicate RREQ.After a time unit has elapsed node 8 receives another RREQ from node 9, containing the path 1-0-9. Node 8 discards this packet as this Request ID exists in the cache and new computed hop count is greater than the stored HopCntMin. Node 8 receives another RREQ from node 4 containing route record as 1-3-4, node 8 discards this RREQ as this Request ID exists in the cache and new computed hop count is greater than the stored HopCntMin. Node 10 is a regular node and it receives three RREQs. Two from node 8 and one from node 9, after receiving first RREQ from node 8 it updates the trust level of the node 0 in its cache, computes the hop count of the path as HopCountMin, stores this value in its cache and adds its own address and rebroadcast the RREQ, after receiving second RREQ from node 8 it updates the trust level of the node 3 in its cache, it compares it with the stored HopCountMin in its cache, as the computed hop count is equal to the stored HopCountMin, it adds its own address and then rebroadcast the duplicate RREQ. After receiving RREQ from node 9 it compares it with the stored HopCountMin in its cache, as the computed hop count is equal to the stored HopCountMin, it adds its own address to the RREQ and then it broadcasts this duplicate RREQ. When node 11 receives first RREQ from nodes 8 having route record as 1-0-8, it updates the trust levels of the nodes in its trust table, computes the hop count of the path as HopCountMin, stores this value in its cache, adds its own address and trust of nodes 0 & 8 and rebroadcast the RREQ, after receiving second RREQ from node 8 it updates the trust level of the node 3 in its trust table, it compares it with the

stored HopCountMin in its cache, as the computed hop count is equal to the stored HopCountMin, it adds its own address and trust of nodes 3 & 8 and then rebroadcast the duplicate RREQ. It drops all the RREQ received from node 10 as hop count of the reverse path to source in RREQ is greater than the stored HopCntMin. When node 12 receives first RREQ from nodes 8 having route record as 1-0-8, it updates the trust levels of the nodes in its trust table, computes the hop count of the path as HopCountMin, stores this value in its cache, adds its own address and trust of nodes 0 & 8 and rebroadcast the RREQ, after receiving second RREQ from node 8 it updates the trust level of the node 3 in its trust table, it compares it with the stored HopCountMin in its cache, as the computed hop count is equal to the stored HopCountMin, it adds its own address and trust of nodes 3 & 8 and then rebroadcast the duplicate RREQ. Node 6 receives three RREQs, two from node 8 and one from node 4, after receiving first RREQ from node 8 it updates the trust levels of the nodes in its trust table, computes the hop count of the path as HopCountMin, stores this value in its cache, adds its own address and rebroadcast the RREQ containing route record as 1-0-8-6, after receiving second RREQ from node 8 it updates the trust level of the node 3 in its trust table, it compares it with the stored HopCountMin in its cache, as the computed hop count is equal to the stored HopCountMin, it adds its own address and then rebroadcast the duplicate RREQ containing route record as 1-3-8-6 and after receiving RREQ from node 4 it updates the trust levels of the nodes, compares it with the stored HopCountMin in its cache, as the computed hop count is equal to the stored HopCountMin, it adds its own address and then rebroadcast the duplicate RREQ containing route record as 1-3-4-6. Node 6 drops the RREQ coming from node 5 as hop count of the route is greater than the stored HopCntMin. Node 12 drops all the RREQs coming from node 6 as the hop count is greater than the stored HopCntMin. Node 4 is a Leader node running watchdog and nodes 6 & 7 share trust tables with node 4. If trust table of node 7 is updated before the RREQ from node 5 reaches to node 7 then node 7 check the trust level of node 5 in its Trust table on receiving RREQ and drops all the data and control packets coming from node 5. Otherwise it computes the hop count, adds its own address to the RREQ and rebroadcast the request containing route record as 1-3-4-5 and later on this request will be discarded by the next leader node 16. Node 15 receives four RREQs, two from node 11 and two from node 12 and it rebroadcasts all the RREQs after updating of its cache and after adding its own address to it as 1-0-8-11-15, 1-3-8-11-15, 1-0-8-12-15 and 1-3-8-12-15. When node 16 receives first RREQ from nodes 12 having route record as 1-0-8-12, it updates the trust levels of the nodes in its trust table, computes the hop count of the path as HopCountMin, stores this value in its cache, adds its own address and trust of nodes 0, 8 & 12 and rebroadcast the RREQ, after receiving second RREQ from node 12 it updates the trust level of the node 3 & 8 in its trust table, it compares it with the stored HopCountMin in its cache, as the computed hop count is equal to the stored HopCountMin, it adds its own address and trust of nodes 3, 8 & 12 and then rebroadcast the duplicate RREQ. Node 16 drops all the other RREQs coming from other nodes as the RREQs received from them have higher hop count than HopCntMin. The destination node 14 receives four RREQs and send RREPs in response to them through reverse route.

5. CONCLUSION

The proposed protocols in [18, 19, 20, 23, 27] fail to work when malicious nodes collude together in order to harm the network. Existing protocols proposed in [14-27] require each network node to work in promiscuous mode. Protocols [19, 24] detect malicious nodes inducing only one or two types of attacks. Protocols proposed in [14, 15, 18, 22] increase delay in route discovery. In [14, 15], proposed protocols result in the most trustworthy path but do not discover the shortest path. Some protocols require each node to have high memory capacity as they store large tables used for storing security information. There are other methods also which are proposed in the literature to improve the routing security [30, 31]. In this paper a new trust based routing protocol TRDSA is proposed which overcomes the above mentioned limitations of the existing protocols in this area.

6. REFERENCES

[1] C. E. Perkins and P. Bhagwat, ”Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,” Comp. Commun. Rev., pp. 234-44, Oct. 1994.

[2] T. Clausen, G. Hansen, L. Christensen, and G. Behrmann, ”The optimized link state routing protocol, evaluation through experiments and simulation,” In IEEE Symposium on Wireless Personal Mobile Communications, Sept. 2001.

[3] C. E. Perkins and E. M. Royer, ”Ad-hoc On-Demand Distance Vector Routing,” Proc. 2nd IEEE Wksp. Mobile Comp. Sys. and Apps., pp. 90-100, Feb. 1999.

[4] D. B. Johnson and D. A. Maltz, ”Dynamic Source Routing in Ad-Hoc Wireless Networks,” Mobile Computing, T. Imielinski and H. Korth, Eds., Kluwer, pp. 153-181, 1996.

[5] L. Zhou and Z. Haas, ” Securing ad hoc networks,” IEEE Netw.., Vol. 13, Issue 6, pp. 24-30, 1999.

[6] M. Zapata and N. Asokan, ” Securing ad hoc routing protocols,” in Proceedings of 3rd ACM workshop WiSE, pp. 1-10, Sep. 2002.

[7] P.Papadimitratos and Z. Haas, ”Secure data transmission in mobile ad hoc networks,” in proceedings of ACM workshop WiSE, pp. 41-50, Sep. 2003.

[8] S. Marti, T. J. Giuli, K. Lai, and M. Baker, ”Mitigating routing misbehavior in mobile ad hoc networks,” in the Proceedings of Sixth Ann. Int’l Conf. Mobile Computing and Networking (MobiCom), pp.255-265, 2000.

[9] S. Buchegger and J. Boudec, ”Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes-Fairness in Distributed Ad Hoc NeTworks,” in the Proceedings of IEEE/ACM Workshop Mobile Ad Hoc Networking and Computing (MobiHOC), pp. 226-236, 2002.

[10] K. Sanzgiri, B. Dahill, B. N. Levine, E. M. Royer and C. Shields, ”A secure routing protocol for ad hoc networks,” In Proceedings of the International Conference on Network Protocols (ICNP), IEEE Press, pp. 78-87, 2002.

[11] Y.C. Hu, D. B. Johnson and A. Perrig,” ‘SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad hoc Networks”, Proc. 4th IEEE Workshop Mobile Comp. Sys. And Applications, pp. 3-13, Callicoon, NY, June 2002.

[12] Y.C. Hu, A. Perrig and D. B. Johnson, ”Ariadne: A secure on-demand routing protocol for ad hoc networks,” In Proceedings of the Eighth Annual International

Conference on Mobile Computing and Networking (MobiCom), ACM Press, pp. 12-23, 2002.

[13] A. Perrig, R. Canetti, D. Tygar and D. Song," The TESLA broadcast authentication protocol," RSA CryptoBytes 5(2), 2002.

[14] T. Ghosh, N. Pissinou, K. Makki. *Collaborative Trust-based Secure Routing Against Colluding Malicious Nodes in Multi-hop Ad Hoc Networks*. 29th Annual IEEE International Conference on Local Computer Networks, 2004, 224 - 231.

[15] T. Ghosh, N. Pissinou and K. Makki. *Towards Designing a Trusted Routing Solution in Mobile Ad Hoc Networks*. Mobile Networks and Applications, Springer Science, 10, 2005, 985-995.

[16] A.A. Pirzada, A. Datta, C. McDonald. *TRUST-BASED ROUTING FOR AD-HOC WIRELESS NETWORKS*. IEEE, 2004, 326-330.

[17] A.A. Pirzada and C. McDonald. *Deploying trust gateways to reinforce dynamic source routing*. Proceedings of the 3rd International IEEE Conference on Industrial Informatics, IEEE Press, 2005, 779-784.

[18] A.A. Pirzada and C. McDonald. *Trust Establishment In Pure Ad-hoc Networks*. Wireless Personal Communications, Springer, 37, 2006, 139-163.

[19] Asad Amir Pirzada, Amitava Datta, Chris McDonald. *Incorporating trust and reputation in the DSR protocol for dependable routing*. Computer Communications, Vol. 29, Issue 15 (5 September 2006), 2806-282.

[20] A.A. Pirzada and C. McDonald. *Dependable Dynamic Source Routing without a Trusted Third Party*. Journal of Research and Practice in Information Technology, Vol. 39, Issue 1 (February 2007).

[21] Xiaoqi Li, M.R. Lyu, Jiangchuan Liu. *A Trust Model Based Routing Protocol for Secure Ad Hoc Networks*. Aerospace Conference, IEEE, Vol.2, 2004, 1286 - 1295.

[22] S. K. Dhurandher and V. Mehra. *Multi-path and message trust-based secure routing in ad hoc networks*. Proc. Int. Conf. Advances in Computing, Control and Telecommunication Technologies (ACT 2009), Trivandrum, India (Dec. 28-29, 2009), 189-194.

[23] X. Li, Z. Jia, P. Zhang, R. Zhang, H. Wang. *Trust-based on-demand multipath routing in mobile ad hoc networks*. Information Security, IET, Vol. 4, Issue 4 (Dec. 2010), 212.

[24] Jian Wang, Yanheng Liu, Yu Jiao. *Building a trusted route in a mobile ad hoc network considering communication reliability and path length*. Journal of Network and Computer Applications, Vol. 34, Issue 4 (July 2011), 1138-1149.

[25] Hothefa Sh.Jassim, Salman Yussof, Tiong Sieh Kiong, S. P. Koh, Roslan Ismail. *A Routing Protocol based on Trusted and shortest Path Selection for Mobile Ad hoc Network*. Communications (MICC), 2009 IEEE 9th Malaysia International Conference on Communications (Dec. 2009), 547 - 554.

[26] Imran Raza, S.A. Hussain. *Identification of malicious nodes in an AODV pure ad hoc network through guard nodes*. Computer Communications, Vol. 31, Issue 9 (Jun. 2008), 1796-1802.

[27] E. Ayday, F. Fekri. *A protocol for data availability in Mobile Ad-Hoc Networks in the presence of insider attacks*. Ad Hoc Networks, Vol. 8, Issue 2 (March 2010), 181-192.

[28] Yamin Li, Shietung Peng, Wanming Chu. *An Efficient Algorithm for Finding an Almost Connected Dominating Set of Small Size on Wireless Ad Hoc Networks*. IEEE, 2006, 199-205.

[29] H. Zafar, D. Harle, I. Andonovic, Y. Khawaja. *Performance evaluation of shortest multipath source routing scheme.* IET Commun, Vol. 3, Iss. 5, 2009, 700–713.

[30] Umang Singh, B. V. R. Reddy, M. N. Hoda. *Enhanced Intrusion Detection System for malicious node detection in adhoc routing protocol using minimal energy consumption*. IEEE Journal Communications IET, ISSN-1751-8628, DOI: 10.1049/iet-com.2009.0616, Volume 4, Issue 17 (Nov. 2010), 2084-2094.

[31] Umang Singh, B. V. R. Reddy, M. N. Hoda. GNDA: Detecting Good Neighbor nodes in adhoc routing protocol. IEEE Second International Conference on Emerging Applications of Information Technology, DOI 10.1109/EAIT.2011.62, 235-238.