Middle-East Journal of Scientific Research 24 (12): 3869-3880, 2016
ISSN 1990-9233
© IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.3869.3880

A Survey on Anonymous Secure On Demand Routing Protocols in MANETs

R. Raghu and T. Menakadevi

Department of IT&ECE, Adhiyamaan College of Engineering/
Anna University, Tamilnadu, India

Abstract: A MANET is a multi hop wireless network with infrastructure less, highly dynamic and frequently changing topology. In MANET nodes are communicate each other without any infrastructure and these generic characteristics of MANET is more vulnerable to security attacks due to highly dynamic environment. Because of nature of MANETs secure routing is challenging task. Newly many routing protocols has been proposed for MANETs. This paper presents a overview and comparison of the on demand protocols characteristics, functionality, anonymity, Unobservability and Unlinkability.

Key words: Routing · Security · Anonymity · Unlinkability · Unobservabilty

INTRODUCTION

Mobile Ad hoc networks are more vulnerable for security measures than the wired networks and several security issues. In MANET all nodes communicate with neighbors, all the nodes works as a router and maintaining a routing tables. And node cooperation is essential in the Mobile Ad hoc networks. In wired networks, one has to get right of entry to wired cables so as to make eavesdrop communications. In contrast, the attacker only needs an appropriate transceiver to receive the wireless signal without being detected. In wired networks, devices like desktops are always in static and not moving from one place to another. Due to this wired networks there is no need to protect users' mobility behavior or movement pattern, due to this sensitive information should kept private from adversaries in wireless environments. Otherwise, an adversary can able to profile users according to the behaviors and cause danger to or harm users based on such information. Lastly, by providing privacy protection for ad hoc networks which has low-power wireless devices and low-bandwidth network connection should be a very challenging task. A number of on demand routing schemes have been proposed for ad hoc networks for recent years and they provide a different level of privacy protection at different cost. With regard to privacy existing on demand routing protocols fails to provide anonymity, unlinkability and unobservability and these features are not achieved fully in existing protocols like AODV, DSR etc. The anonymous secure routing protocols stated in this paper aims to offer the following privacy properties.

Anonymity: The senders, receivers and intermediate nodes are not identifiable within the whole network or in the largest anonymity set.

Unlinkability: Is the connection between senders and receivers, the intermediate nodes and the messages is to protected from the outsiders. Node linkage between two messages, e.g., whether they are from same source node, are also to protect.

Unobservability: Any significant packet in the routing pattern is indistinguishable from other packets to an external attacker. And not only the content of the packet and also the packet header similar to packet type are secured from eavesdroppers. And if any node involved in route discovery or packet forwarding, it includes the source node, destination node and the intermediate node, should not be aware of the identity of other involved.

Related Work: On demand protocols generate routes only when source node need to send packets to destination for transmission. When a source requires a route to destination, it will initiates route discovery process within the network. This process is accomplished once a route is to be found or all possible route

Corresponding Author: R. Raghu, Department of IT&ECE, Adhiyamaan College of Engineering/Anna University, Tamilnadu, India.

permutations are examined. Once a route is discovered and constructed or established, it is preserved by the route maintenance procedure until or either destination becomes unreachable along every path from source or route is no longer to be get desired.

Ad hoc On-demand Distance Vector Routing protocol (AODV) [1] is an enhancement of the DSDV algorithm. AODV reduces the number of broadcasts by generating on-demand routes as opposite to DSDV that keeps the list of all the routes. In AODV, when a node *'S'* wants to send a packet to destination node '*D'* and does not have a route to *'D'*, AODV initiates route discovery by broadcasting a RREQ to its neighbors. The immediate neighbors node which receive this RREQ rebroadcast the same Route request to their neighbors. This method is repeated until the Route request reaches the destination. After receiving the first reached RREQ, the Route Reply was sent back to source by destination node through the opposite path where the RREQ arrived. The destination node will ignore the same RREQ that arrives late. In addition, AODV allows intermediate nodes that have sufficiently fresh routes (with destination sequence number which are equal or greater than the one in the RREQ) to generate and send an RREP to the source node.

The key feature of DSR [2] is source routing, which means the sender knows the complete node by node route to destination. The node keeps route caches containing the source routes that it is attentive of each node updates entries route information in the route cache as and when it studies about fresh routes. The data packets carry the source route information in the packet headers. Delay and throughput forfeits of DSR are mainly credited to insistent use of caching and lack of any mechanism to detect expired spoiled routes or to determine the freshness of routes when few choices are on hand. Aggressive caching helps in DSR at low loads and also keeps its routing load down.

Destination Source Routing (DSR) [3] uses the MACs and shared keys between nodes to authenticate between the nodes and uses time stamps for packet lifetime. Wormhole attacks are possible in Ariadne between two compromised nodes. DSR prevents spoofing attack with time stamps. The use of source routes stops loops, since a packet travelling through only authentic nodes will not be sent into a loop due to time stamps.

Secure routing protocol (SRP) [4] was introduced and constructed based on Destination Source Routing (DSR). The nodes called intermediate nodes participating in the route discovery measures the frequency of queries received from their neighbors and preserve a priority ranking inversely proportional to query rate. So the malicious cooperated participating nodes in the network are given least priority to deal with secure routing. The security scrutiny is analogous to Ariadne as it was based on DSR protocol.

Secure Ad hoc On-Demand Distance Vector Routing (SAODV) SAODV [4] is a widely implemented protocols in communication engineering due to its robust security features. SADOV utilizes central key management in its routing topology. Digital signature mechanism is used to validate at node level and hash chain is used to prevent the altering of node counts by using password protection mechanism. Tunneling attacks are possible between two compromise nodes. Wormhole attacks are forever possible with cooperated nodes in any ad hoc network topology. Sequence numbers is used to prevent most of the possible replay attacks.

Existing Protocols

Anonymous Secure on Demand Routing Protocols

Anonymous Secure Routingprotocol (ASR): Anonymous Secure Routing protocol (ASR) [5] provides additional properties on anonymity, i.e. Node Identity anonymity and Strong Location Privacy and at the same time it ensures the security of exposed routes against various passive and active attacks. ASR can achieve both anonymity and security properties. The aim of ASR is protecting the privacy of nodes and routes and at the same time ensures the security of discovered routes. Ensuring Privacy includes two types i.e

- Identity Privacy: Identity Privacy has following requirements:
    - No one in network knows the identities of the source and the destination, other than themselves.
    - The source and the destination nodes have no information about the real identities of intermediate nodes on the route.
- Location Privacy: It has following requirements:
    - No one in network knows the exact location of the source or the destination, other than themselves.
    - Other nodes, typically intermediate nodes on the route, have no information about their distance, i.e. the number of nodes, from either the source or the destination. This requirement is optional, but it is attractive in keeping both node identity

and location anonymity identity of the source or the destination, exclusively when the distance is one hop.

For a protocol satisfying (a), protocol provides Weak Location Privacy; if a protocol satisfying both (a) and (b), we can say that such protocol offers Strong Location Privacy. Here ASR achieves Strong Location Privacy. The ASR protocol consists of the following parts: RREQ, RREP, Anonymous Data Transmission and Route Maintenance.

Route Request:

$$[RREQ, Seqno, K_T(\, dest, K_s, U_0), K_s(Seq, END),] \tag{1}$$

$$[Ps = (\, K_{max+1}).P).\ P_x \tag{2}$$

In the period of the route request process, each node in route denoted as X_i (i = 1, 2, 3, 4, 5,... . ,n) receives a request of route as like below format:

Let K_{max} denote the maximum number of nodes that Source wish the route to be. Then, For instance, ‘X_i ‘ chooses the length of the random number, i.e. ‘S_i‘, is 16, source nodes wants to discover a route between the destination and the node itself and expect the length of the route is no more than 10 nodes. According to Equation (2), the ‘P_x‘ is 176 and thus generate a random number ‘U_0’ among 176 bits during the generation of the route request message. When receiving the route request packet, each forwarding node denoted as ‘X_i’ first checks whether seqno is recorded in its route table. If seqno is available, it discards the packet without decrypting the third element of the RREQ. Otherwise, ‘X_i’ tries to decrypt “K_T $(dest, K_{,s}\ U0)$”. If fails, ‘X'_i records the seqno, generates ‘U_i’ and then replaces ‘$U\ '_{i-1}$ with ‘U'_i respectively. Finally, ‘X_i’ broadcasts the customized packet locally. If succeeds, it means that ‘X_i’ act as an destination node of this packet, because only the destination can fruitfully decrypt the packet. Afterwards, node ‘D’ compares ‘U_0’ it is recovered from the third element of the Route request packet, with Un to recuperate the length of the route, if the length is equal to or less than ‘H_{max}’, then the destination node discards those packets whose Un has been modified by more than ‘H_{max}’ nodes. Thereafter destination node generates and broadcasts a RREP packet for each route with less than ‘H_{max}’ hops.

Route Reply: During the route response process, each node in route denoted as X_i (i = 1, 2, 3, 4, 5,. . . , n) receives a reply of route as like below format:

$$[RREP, \{T_{i+1}\}PK_i, T_{\ i+1}(seq, K'_s\,)] \tag{3}$$

Once receiving the Route reply packet, each forwarding node represented as ‘X_i’ first tries to decrypt “$\{\ T_{\ i+1}\}PK_i$” and recovers the last element of the route reply packet. Since the second element is encrypted by “PK_i” and only ‘X_i’ can decrypt it. Then ‘X_i’ extracts seq from the recovered content and checks whether seq has been logged in its route table. If no, it humbly discards the packet without any checking. Otherwise, ‘X_i’ extracts ‘K_s’ from the recovered information. Thereafter, ‘X_i’ also needs to make positive that the route reply packet is from the destination node. After successfully validating the validity of the route reply packet, ‘X_i’ chooses a random number ‘T_i’ and adds ‘T_i’ and ‘T_{i+1}’ into the record with the corresponding seq. Finally, ‘X_i’ broadcasts the modified RREP packet locally.

Anonymous Data Transmission: Any sending or forwarding node broadcasts packet to its neighbors, after this the neighbors verify the validity of TAG. If the packet passes successfully verification, forwarding node re-calculates and replaces TAG. In count, previous to broadcasting the packet to its neighbors, the content of data packets shuffled by an effective encryption so that the malicious cannot match payload information to trace data forwarding. Suppose packet fails to pass the verification, it is discarded. Such method is repeated until the packet reaches destination node.

Route Maintenance: Nodes can detect route failures when transmission count exceeds a predefined number. While detection, a node looks up the consequent entry in its forwarding table, finds the TAG contents that it shares with the previous node and assembles a route error packet the format: [RERR, TAG].

Efficient Anonymous Dynamic Source Routing for Mobile Ad-hoc Network (ANON DSR): The Anon DSR [6] had proposed to provide three levels of security protection. This novel routing consists of three protocols. The first protocol is used to create a common secret key and random nonce between the source and destination for safe and anonymous communications. The second protocol uses the common secret key and nonce to create a trapdoor and make use of an anonymous onion routing between the source and destination. To provide last level protection the last protocol is used, were session keys are shared with the intermediate nodes of *‘S’* and *‘D’* uses cryptographic onion method to encrypt all communications.

Efficient Anonymous Dynamic Source Routing: Anon DSR includes three protocols they are security parameter establishment, route discovery protocol and data transfer protocol.

Security Parameter Establishment Protocol: This protocol is used to set up the security parameters for safe and anonymous communications using secure parameters type in the packet. This protocol includes two stages: *Route request* stage and *Route reply* stage.

Route Request Phase: The source first creates the below RREQ packet and broadcasts the packet locally

$$<RREQ, Sectype, Seqnum,ID_{src}, ID_{dest}, RRece, Secparam>. \quad (4)$$

where “*Sectype”* parameter represents a security type of the route request packet, *“seqnum”* is sequence number. *“*ID_{src}*”* is identity of the source and *“*ID_{dest}*”* is identity of the destination nodes. *“RRece”* is the source route record and *“SecParam”* contains the security parameters that the source node provides. If *“SecType”* is secure, *”SecParam= { (Ndest PK E K, K, Param),* $Sign_{src}$*}”* where *‘*N_K*’* is a secret key of the shared secret key *‘*K_{Para}*’* is the cryptographic information such as algorithm of encryption and version used in and *“Signsrc”* is a group signature signed by the source node for verification. Every node rejects the same packet when it receives the second time by matching the sequence number and packet type. If it finds the route request packet is a safe or anonymous packet, then it needs to decrypt the *“SecParameter*” first, proves whether the packet is correct, records the information like $<NK, ID_{src}, K, Para >$ into its shared secret keys ring and jumps into the route reply phase.

Route Response Stage: The destination broadcasts the following Route reply packet locally to respond the RREQ packet

$$< RREQ, SecType, Seqnum,ID_{src}, ID\ _{dest}, R_{Rec}, Secpara> \quad (5)$$

Anonymous Route Discovery Protocol: This protocol establishes an anonymous route between source and destination nodes with common secret key and secret key index in their key ring. This protocol includes 2 stages: *Route request* phase and *Route reply* phases

RREQ Phase: The following ANON - route request packet and broadcasts the packet locally

$$<ANON\text{-}RREQ, PK_{temp}, trdest, onion> \quad (6)$$

where ANON-RREQ represents a ANON-RREQ packet type*, “*PK_{temp}*”* is a one-time used temporary public key created by the source, also works as a unique sequence number in the route request phase and *“trdest”* is trapdoor information that only the destination can open with a shared secret key. And Onion routing is also called cryptographic onion message that registers the anonymous path with security protection. It then create a single route pseudonym *“Nx”*, encrypts the pseudonym and received path discovery onion jointly with the session key and broadcasts the fresh packet locally.

RREP Phase: The destination node decrypts the protected onion of cryptographic in the ANON-RREQ using the private keyof its own and confirms if all data are correct. Finally, the destination makes the below ANON-Route reply packet and broadcasts it locally

$$< ANON\text{-}RREP, N_D, PRO_D> \quad (7)$$

All nodes check if *‘*N_D*’* received is their pseudonym after receiving ANON-route reply packet. They discard the packet if ‘N_D*’* is not their pseudonym. Clearly, node *‘D’* can find it is on the route. Then *‘D’* decrypts one layer of the onion “PRO_D“ using its session key “K_D” After the protocol has completed execution, the source node and destination node have updated their shared secret key and key index for the afterward anonymous communication use. And they also have all route pseudonyms and session keys for the current anonymous communication use. In this protocol, cryptographic onion is used to protect the anonymous route record in the ANON-Route request packet and ANON-Route reply packet. In count, every intermediate forwarding node has one public key encryption and one symmetric encryption of key for constructing "*PDO*", one symmetric key decryption for opening the trapdoor in the RREQ phase and one symmetric key decryption for decrypting one level of the onion in the Route reply phase.

Anonymous Data Transfer Protocol: This Protocol builds a onion of cryptographic for anonymous communication data security. This protocol is used when an anonymous route discovery protocol is completed. The protocol is described as follows. Source node creates a onion of cryptographic for the communication data that the source wants to send to the destination, creates the below ANON-DATA packet and broadcasts the packet locally

$$< ANON\text{-}RREP, N_{src,} onion > \quad (8)$$

Each intermediate forwarding node checks whether the pseudonym of the packet belongs to it and decrypts one layer of the onion using its session key if it is on the anonymous route. Then it changes the route pseudonym by using its forwarding routing table, uses the decrypted onion instead of the expected onion and broadcasts the new packet locally. It rejects the packet if it is not on the anonymous route. This process is repeated until the data packet arrives at the destination. A opposite anonymous data transfer from the destination to the source uses the reverse data onion (RDO).

C.anonymous Location - Aided Routing in Suspicious Manets (ALARM): In many traditional mobile network scenarios, the nodes establish a communication on the basis of persistent public identities. However, in some hostile and suspicious a MANET settings, node identities must not be exposed and the node movements must be not traceable. Instead, nodes need to communicate on basis of nothing more than their current locations. The ALARM [7] is proposed and it uses nodes current locations to construct a secure MANET map. Based on the current maps, each node can decide with other nodes it wants to communicate with hostile node and node identities must not be revealed. ALARM introduced to provide secure communication in hostile and suspicious MANETs.

ALARM: Anonymous Location-aided Routing in MANETs: ALARM requires off-line Group Manager (GM) that initializes the underlying group signature scheme and enrolls all valid MANET nodes as group members. In case of a dispute, the *"GM"* is responsible for opening the contested of group signature and determine the signer. Depending on the specific Group Signature scheme, the *"GM"* may also have to handle future join for new members as well as revocation of existing members. The operations of *"ALARM"* is shown below

- Time is divided into time slots duration *'T'*. At the beginning of every slot, each node broadcasts a message containing a location (GPS coordinates), time-stamp, temporary public key and a group signature computed over these fields. This is called Location Announcement Message (LAM). Each *"LAM"* is flooded throughout the MANET. In the period between successive *"LAM-s",* a node can be reach using a pseudonym which is set to the group signature in its last *"LAM"*. (Assuming, of course, the signature is valid.) Each node receives a *"LAM",* first verifies the group signature.
- If the signature is valid, the node communicates the message to its neighbors unless it has previously received the same message. Having collect a current *"LAM-s",* each node can easily construct a geographical map and a connectivity graph of the MANET. If a node need to communicate to certain location, it first checks to see if there is a node at (or near) that location. If so, it sends a message to the target pseudonym.

Privacy Preserving Location Based on Demand Routing in MANETs: The on-demand location-based anonymous MANET routing protocol (PRISM) [8] is proposed to achieve privacy and security in contradiction of both outsider and insider adversaries. If the current MANET topology is unknown and there are no long-term node identities, one possibility is to use a hit and miss approach, in it, a node picks a geographical location coordinates, draws certain perimeter around it (e.g: by specify a radius or points of a polygon) uses the resultant area as the destination addres. The message (route request) addressed in such the way propagates through the network (via flooding, as in AODV) and either fails to find all nodes in the specified area or reaches one or more. Destination node(s) then reply (if they want to) using states along the reverse route, with intermediate nodes using information cached during route requested processing. This location-based method is effective as it guarantees that, as long as the network is connect, all destination within the specified area are reached.

PRISM Protocol: PRISM allows a source to specify a destination area and simultaneously discover a multiple destination nodes in it.

- The source broadcasts a route request (RREQ) it contains the destination locality, in the form of coordinates and a radius – Destination- AREA. RREQ also contains temporary public key *"PKTMP"*, a time-stamp *"TSSRC"* and a group signature, *"GSIGSRC"* computed over all previous fields. Note that the source starts searching in an area with a smaller radius and if no reply received within a specific time window, it increases the radius the area and sends another *"RREQ"*. A received *"RREP"* considered erroneous if time-stamp is incorrect, the exact location of the replying node is not within the destination area or the verification of group signature included in the *"RREP"* fails.
- Upon receiving a *"RREQ",* each node first checks *"TSSRC"* is valid. If not the route request is dropped. Next, a node checks whether it has previously processed the same *"RREQ"*. This can be done by computing a hash of the new Route request *"H(RREQ))"* and looking up in the local cache where recently handled *"RREQ"* hashes are stored. Then, the node checks whether it's within *"DST-AREA:(A)"* If not, the intermediate node caches *"H(RREQ)"* and re-broadcasts the *"RREQ"*. If the node within the destination area, it verifies *"GSIGSRC"*. If invalid, the Route request is discarded. Otherwise, it stores the entire route request containing *"GSIGSRC"*. The destination then composes a route reply (RREP) which contains: *"(H(RREQ)",* a new random session key *'K_S'* and the exact destination location. Both are encrypted under *"PKTMP"* obtained from the route request. The route reply also includes the group signature *"GSIGDST"* for all fields. Lastly, the destination broadcasts *"RREP"*.
- Upon receiving a Route reply, every node checks whether has cached the corresponding *"H(RREQ)"*. If not, the Route reply is dropped since this node was not on the forward route. *"H(RREQ)"* is already cached, the node checks if the same Route reply has been processed. If so, the route reply is dropped. Then the intermediate node now generates a new entry in its dynamic route table and re-broadcasts the *"RREP"*. Each active table entry must contains: *"H(RREQ)", 'H'* (Route reply) and the time-stamp of entry creation. When the Route reply is received, the source node initially first checks for the exactness of the time-stamp and the accurate location of the replying node then verifies the group signature if it is Invalid, the *"RREP"* is discarded and logged as a failure. Next, the source decrypts the session key and location supplied the destination. This key is consequently used for message encryption and/or authentication. This completes the route set-up method. Once the route is established, each source-destination data message specifies the tuple of route request and route reply hashes*,"< H(RREQ),H(RREP) >",* as unique route identifier. In the reverse direction, the reverse tuple "*<H(RREP),H(RREQ) >"* is used a route identifier.

ANDOR: an Anonymous on Demand Routing Protocol with Untraceable Routes for Mobile Adhoc Networks: ANODR, anonymous on-demand routing protocol which is used for mobile ad hoc networks and to be deployed in hostile environments. For route anonymity, ANODR[9] ensures that adversaries cannot discover the real identities of a local transmitters. The plan of ANODR is based on broadcast method with a trapdoor information, namely "broadcast" and "trapdoor information". An ANODR divides the routing process into two types: anonymous route discovery and anonymous route maintenance.

Anonymous Route Discovery: Anonymous route discovery is important procedure that can establish a random route pseudonyms for on demand route. A communication source must initiates the route discovery process by assembling an *"RREQ"* packet and it may be locally broadcasting it. The *"RREQ"* packet is shown below.

<RREQ, Seqnum, trdest, onion> (9)

ANODR-TBO: Anonymous Route Discovery - Trapdoor Boomerang Onion: When intermediate forwarding node *'X'* receives RREQ packet, it broadcasts the RREQ locally. The boomerang onion will bounced back by the destination that look Like the public key version, when node X sees the RREP packet, it strips a layer of the boomerang onion and broadcasts and it modifies the RREP packet. Finally the source will be receiving the boomerang onion originally sent it out. Then the trapdoor is made in the RREQ phase, hence the result is equal to the wireless unicast routing. Then the node strips a cover of the boomerang onion and broadcastes the customized RREP packet.

Anonymous Data Forwarding: For each end-to-end connection the source node wraps its data packets by using the outgoing route pseudonym in its forwarding table. Then the data packet is broadcasted locally without identifying sender and the local receiver. The sender does not worry to react to the packet sent out. Remaining all the other locally receiving nodes must look up the route pseudonym in their dispatching tables. Then the node discards the packet if no equivalent is found then it returns back to it. Otherwise, it will change the route pseudonym to the matched outgoing pseudonym, then it will broadcasts the changed data packet made locally. Then the procedure is repeated until the data packet arrives at the destination end.

Anonymous Route Maintenance: The routing table entries are recycled upon its timeout *'T'*. In addition, when one or more hop is broken due to mobility, nodes cannot forward packet via the broken hops. Nodes can detect such anomalies when the re-transmission count exceeds. Upon this anomaly detection, a node looks up the consequent entry in its forwarding table, Then it finds the other route pseudonym *'N'* which is connected with the pseudonym *'N'* of the out of order hop and assembles a route error packet of the format *"<RERN>"*. Then the node recycles the table entry and locally broadcasts the *"RERR"* packet.

Proposed Protocol

Enhanced Efficient Secure Un Observable On Demand Routing Protocol (EESUOR): The EESUOR protocol is proposed to achieve information unobservability and anonymity of all types of packets.

An ad hoc network consisting of '*n' nodes*, all *'n'* nodes have the same communication range and each node can move around within the network. Each node in network can communicate with other nodes within its transmission range and these nodes are called its neighbors. For nodes outside transmission ranges have to communicate via a multi-hop path. Assume that the ad hoc network is well connected and each node contains at least one neighbor.

Before the ad hoc network starts up, make sure that all nodes in topology in promiscuous mode. In promiscuous each node in the network listens the packets transmitted to other nodes and also listens the packets received from other nodes. Each node in topology send beacon messages to other nodes to keep track of its neighbors and to calculate trust value in node. By evaluating the trust computation formula the trust value is calculated in all nodes and stored in routing table of particular nodes.

By initiating group signature scheme, a group public key "**gpk"** is gererated and it is publicly known by every node in legitimate group and it also generates a private group signature key $\mathbf{gsk}_X$ for each node called *'X.'* The group signature scheme ensures full-anonymity, which means a signature does not disclose the signer's uniqueness but everyone can verify its validity in within group. Group signature scheme provides anonymity facility to legitimate group members. Anonymous key establishment process is applied on nodes to construct a set of session keys with each of its trusted reserved neighbors. As a result of this phase, a pair wise session key is constructed anonymously, in this way the two nodes establish this key without knowing who the other party is.

After completing the anonymous key construction process, privacy-preserving route discovery process is take place based on the keys established and *"trust req"* in previous phase. Similar to normal route discovery process, privacy-preserving route discovery process consists route request and route reply. Under the protection of session keys and *"trust req"* the route discovery process can be started by the source node to discover a route to the destination node.

Once the route request and route reply is successfully completed means and if source node '*S'* successfully finds out a route to the destination node *'D'*, then '*S'* can start unobservable data transmission under the protection of pseudonyms and keys.

The proposed protocol has four phases that are involved in the process of secure routing. They are Establishment of Trust Based intermediate nodes, Anonymous Key establishment, Privacy-Preserving Routing and Unobservable Data Transmission.

Routing Procedure of Eesuor: The routing algorithm is implemented based on existing on- demand "USOR" protocol. The main routing procedure can be summarized as follows. (Fig. 1)

- Before route discovery process trust cumulative value is calculated and stored in its routing table by evaluating formula of (10).
- During route finding process, a source node called *'S'* broadcasts an route request in the format of (11).
- If intermediate node *'X'* receives route request, it try to verify 'E_D' part present in the route request by using its group public key.
- If the verification is successful it acts as a destination otherwise node *'X'* act as an intermediate node.

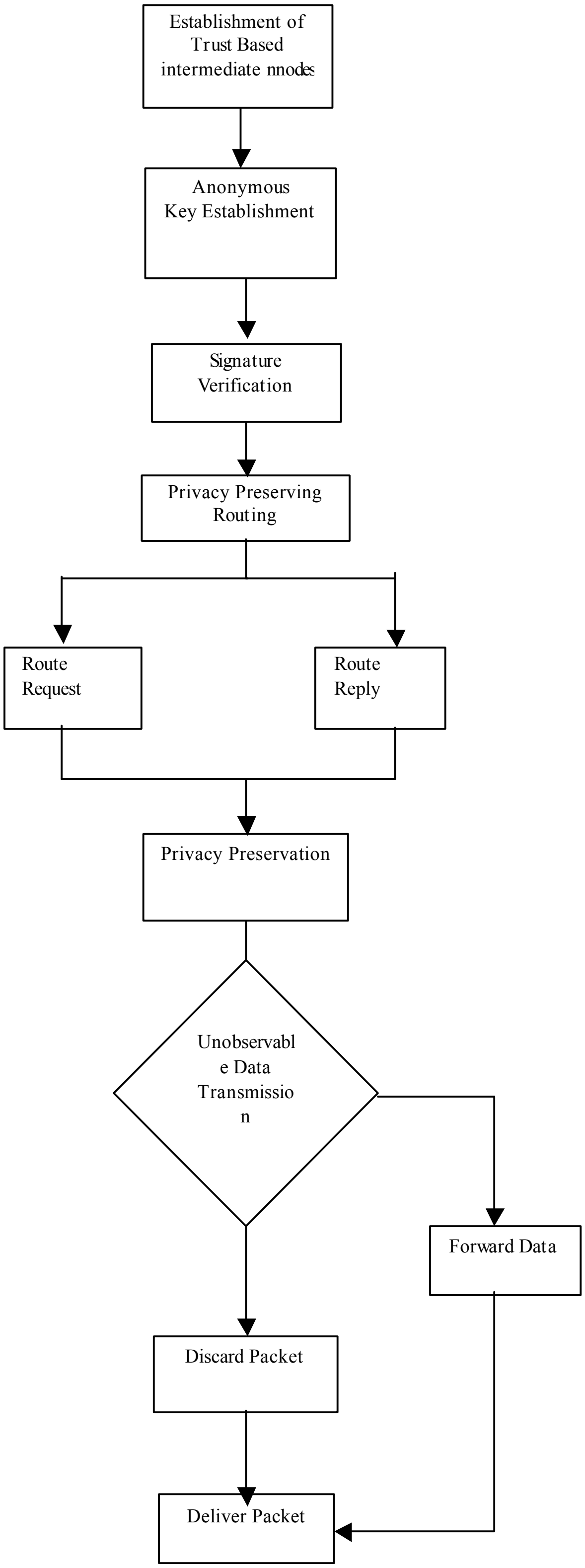
Establishment of Trust Based intermediate nnodes
Anonymous Key Establishment
Signature Verification
Privacy Preserving Routing
Route Request
Route Reply
Privacy Preservation
Unobservabl e Data Transmissio n
Forward Data
Discard Packet
Deliver Packet

Fig. 1: Enhanced Efficient Secure Un Observable On Demand Routing Protocol

- If node *'X'* is intermediate node then it creates "Nonce" and new pseudonym N_A for *"RREQ"*. Then it broadcasts newly constructed *"RREQ"* to its neighbors.
- Other intermediate nodes do the same as '*X*' does. Finally, the destination node receives the message in the format of (13)
- Once the route request is received by destination then it performs verification, if verification is successful then destination assembles route reply in the format of (14)
- When intermediate node receives *"RREP"* from destination then intermediate node identifies who the sender of the message is by evaluating the *"Nym"*.
- Intermediate node uses the right key to decrypt the cipher text and then intermediate node find out which route this *"RREP"* is related to according to the route pseudonym and seqnumber.
- At the end, Intermediate node chooses a new nonce and computes *"Nym"* and sends the following message in the format of (15).
- Other intermediate nodes perform the same operations as previous intermediate node does. In this way finally the *"RREP"* reaches the source.
- Then Unobservable data transmission takes place after completing the route reply phase successfully.

Establishment of Trust Based Intermediate Nodes: The major objective of this system is to construct trustworthy intermediate nodes and enabling these trusted nodes to participate in the path construction.

In this approach, the trust level in a node is defined as a cumulative value that is based on trust computation formula.

Trust Cumulative Value = Packets Received / Packets Sent (10)

By evaluating above computation formula the trust is calculated in all nodes and trust computation value is stored in routing table of particular nodes.

Anonymous Key Establishment: Nodes in group employs anonymous key establishment to anonymously construct a set of session keys with each of its neighbors. Here Group signature scheme is used for anonymous key establishment. Assume there is a node called '*S*' with private signing key *"gS_k"* and a private ID-based key *'K_S'* in network and it is bounded by a number of neighbors within its communication range. Then in anonymous key establishment procedure, node '*S*' does the following:

- *'S'* generates a random number *'R_S'* and computes a signature of *'R_S'* using its private signing key *"gsk_S"* to obtain "*SIGNgsk$_S$ (R_S)*". Anyone can verify this signature using the group public key *"gp_k"*. It broadcast *"SIGNgsk$_S$(R_S)"* within its neighborhood.
- *'A'* neighbor '*X'* of *'S'* receives the message from *'S'* and verifies the signature in that message. If the verification is successful, *'X'* chooses a random number *'R_X'* using its private signing key *'gsk $_x$ '* to obtain *"SIGNgsk$_x$ (R_x)"*.. Then *'X'* computes the session key *"$K_{SX}=H_2(R_SR_x)$"* and replies to *'S'* with message *"$E_{kSX}(K_X{}^{\star}|Rs|Rx)$) "* where *"$K_X{}^{\star}$ "* is *"X's"* local broadcast key.
- By receiving reply from '*X*', *'S'* validates the signature inside the message. If the signature is valid, '*S*' continues to compute the session key between *'X'* and itself as *"K_{SX} =H_2(R $_S$ R $_x$)"* and '*S*' also generates a local broad-cast key *"$K_s{}^{\star}$"* and sends its neighbor ' *X'* to inform to '*X*' about the established local broadcast key.
- *'X'* receives the message from '*S'* and computes the same session key as *"$K_{SX}=H_2(R_S\ R_x)$"* It then decrypts the message to get the local broadcast key *"$K_S{}^{\star}$"*.

Privacy Preserving Routing: The privacy preserving routing scheme comprises of two phases: route request as the first phase and the route reply process as the second phase. Under protection of session keys, the route request process can be initiated by the source to discover a route to the destination node.

Route Request (RREQ): Source chooses a random number *'R_S'*, identity of node Destination to encrypt a trapdoor information that can only opened with Destination's private ID-based key and selects trust requirement which yields *"E_D(S,D, Trust$_{req}$, R_S)"*.Source then selects or chooses a sequence number *"seqnum"* for this route request and another random number "R_{S1} ", as the route pseudonym, it is used as index to a specific route entry. Source chooses a nonce *"Nonce$_S$"* and calculates a pseudonym as *"Nym$_S$=H3(k $_S{}^{\star}$|Nonce $_S$)"*. After this, Source encrypts these items using its local broad-cast key *"$K_S{}^{\star}$"* to obtain *"$E_{KS}{}^{\star}$(RREQ,R_{S1},E_D(S,D, Trust$_{req}$, R_S)"*.Finally, Source broadcast the following unobservable route request to its neighbors:

Nonce$_S$, Nym$_S$, $E_{KS}{}^{\star}$ (RREQ, R_{S1}, E_D(S,D, Trust$_{req}$, R_S), seqnum).

(11)

By receiving the route request message from Source, node '*A* ' tries all its session keys shared with all neighbors to calculate *"$H_3(K_{XA}|Nonce_S)$"* to see which one matches the received *"Nym$_S$"* Then node '*A'* would find out *"$K_S{}^{\star}$" to* satisfy "Nym$_S$" So *'A'* uses '$K_S{}^{\star}$' to decrypt the cipher text. After identifying this is a route request packet, node '*A'* tries to decrypt *"E_D(S,D, Trust$_{req}$, R_S)"*. Here '*A'* have to verify all information which is present in 'E_D' if it is not the destination and if trial fails, so *'A'* acts as an intermediate node. '*A'* creates a nonce *"Nonce$_A$ "* and a new route pseudonym *"N_A"* for this route. *'A'* then calculates a pseudonym *"Nym$_A$=H3($K_{A*}{}^{\star}$ Nonce$_A$)"*. Node *'A'* also records the route pseudonyms and sequence number in node *'A'* routing table for purpose of routing. At the end, node '*A'* constructs and broadcast the following message to all its neighbors:

Nonce$_A$, Nym$_A$, $E_{KA}{}^{\star}$ (RREQ, R_{A1}, E_D (S,D, Trust$_{req}$, R_S), seqnum).

(12)

Other intermediate nodes do the same as '*A'* does. Finally the destination node receives the following message from C:

Nonce$_C$,Nym$_C$, $E_{KC}{}^{\star}$(RREQ,R_{C1},E_D(S,D, Trust$_{req}$, R_S), seqnum).

(13)

After decrypting the cipher text using *'$K_C{}^{\star}$'*node '*D'* accounts route pseudonyms and the sequence number into its route table. Then '*D*' successfully decrypts *"E_D(S,D, Trust$_{req}$, R_S)"*

Route Reply (RREP): If node '*D'* is the destination node, then '*D*' starts to prepare a reply message to the source node. Node '*D*' chooses a random number *'R_D'* and computes a cipher text *"E_S(D,S, R_D)."* Showing that *'D'* is the legitimate destination able of opening the trapdoor information. A session key *"K_{SD}"* is computed for data protection. Then he generates a new pair wise pseudonym "Nym$_{CD}$" between '*C*' and '*D*'. At the end, using the pairwise session key "K_{CD}", node *'D'* computes and sends the following message to *'C' :*

$$Nonce_D, Nym_{CD}, E_{KCD}(RREP, N_C, E_S(D, S, R_S, R_D),seqnum) \quad (14)$$

When '*C'* receives the above message from node '*D'*, then '*C'* identifies who the sender of the message is by evaluating the equation "Nym$_{CD}$"$_.$ So node *'C'* uses the right key *"K$_{CD}$ "* to decrypts the cipher text and then *'C'* find out which route this RREP is related to according to the route pseudonym 'N$_C$' and seqnum. At the end, '*C*' chooses a new nonce "Nonce$_C$", computes "Nym$_{BC}$" and sends the following message to *'B'*:

$$Nonce_C, Nym_{BC}, Ek_{BC}(RREP, N_B, E_S(D, S, R_S, R_D), seqnum). \quad (15)$$

Other intermediate nodes perform the same operations as '*C'* does. In this way finally the RREP reaches the source.

Unobservable Data Transmission: The Unobservable data transmission takes place after completing the route reply phase successfully. The packets from *Source* must traverse '*A'*, '*B'* and node '*C'* to reach '*D'*. The data packets sent by *Source* takes the following format

$$Nonce_S, \mathrm{Ny}_{SA}, E_{kSA}(DATA, N_S, seqnnum, E_{kSD}(payload)) \quad (16)$$

Upon receiving the message from *Source,* node *'A'* composes and forwards the following packet to node '*B'*:

$$Nonce_A, \mathrm{Ny}_{AB}, E_{kAB}(DATA, N_A, seqnonum, E_{kSD}(payload)). \quad (17)$$

The data packet is forwarded until it reaches the final or destination node '*D'*. At the end, the below data packet is received by '*D'*:

$$Nonce_C, \mathrm{Ny}_{CD}, E_{kCD}(DATA, N_C, seqnum, E_{kSD}\ (payload)) \quad (18)$$

Comparison: The important difference between ANODR, EESUOR and AnonDSR is, the protocol EESUOR trusts on established keys between nodes to attain privacy protection. But the other two protocols depends on onion encryption and end-to-end security. Subsequently, EESUOR can provide complete Anonymity, unobservability and unlinkability well, but AnonDSR and ANODR flop to protect unobservability or linkability of messages. ALARM protocol is based on Location Announcement Message (LAM) so there is chance for leaking Location of nodes. ALARM can't achieve well Anonymity, unobservability and unlinkability. In PRISM the source broadcasts a route request (RREQ) it contains the destination locality, in the form of coordinates and a radius – Destination-AREA and PRISM fails to offer bothAnonymity and unobservability. All protocols mentioned in Table 1 releases some valuable information related to RREQ and RREP but EESUOR doesn't release any information while performing route request and route reply. DSR is based on source routing the route request packets travels in single path established by source and route reply packets travels in reverse path so due to mobility if any node moves out of range retransmission get failure this can be overcome with EESUOR and EESUOR is not based on source routing.

Table 1 shows Comparison of Routing Protocols with respect to information known by source and information present in RREQ and RREP.

Table 2 summarizes parameters which knows by source while initiating route request. And Table 2 also summarizes parameters used in RREQ and RREP of different protocols.

Routing protocols are specially designed to work in different MANETs environment or scenarios [11]. The protocols PRISM and ALARM is developed for Location Based routing or for location based anonymous communications. The topology based protocols like Anon DSR and ANODR contains problems in providing unobservability and unlinkability. ANDOR concentrations on protecting the node identity or route identities during discovery of routes. Global trapdoor information is used in ANDOR route discovery process instead of utilizing destination node ID.Route Pseudonym and Ids of nodes in neighborhood is probably visible in Anon DSR. The Protocols PRISM and EESUOR include

Table 1: Observable Information of Routing Protocols

Protocol	Observable Information
ASR	Sequence no., trapdoor info., RREQ/RREP tag
Anon DSR	Trapdoor info., RREQ/RREP tag
ALARM	RREQ/RREP tag, Location
PRISM	RREQ/RREP tag, Location
ANODR	Sequence no., trapdoor info., RREQ/RREP tag
EESUOR	No Observable Information

Table 2: Info. known by source and info present in RREQ, RREP& RREP

Protocol	Information Known by Source	Information in route discovery	Information in route Reply
ASR	Dest ID of node	Random number, No of Hops, Dest ID	Seqno, Random number
Anon DSR	Public key, Dest ID of node.	Destination Trapdoor	Source Trapdoor
ALARM	ID of Source	location (GPS coordinates), time-stamp, temporary public key	Time-stamp, temporary public key
PRISM	Public key of group	Destination Area	Destination location
ANODR	Public key and Trapdoor info of Destination	Route Pseudonym	Route Pseudonym
EESUOR	Public key and ID of destination	Destination Trapdoor information	Source Trapdoor information

Table 3: Authentication & Topology Type used in Protocols

Protocol	Authentication	Category
ASR	None	Topology Based
Anon DSR	None	Topology Based
ALARM	Group Signature	Location Based
PRISM	Group Signature	Location Based
ANODR	None	Topology Based
EESUOR	Group Signature	Topology Based

authentication methods to sign the packets. EESUOR uses group signature authentication mechanism for signing routing packets.

CONCLUSION

This paper discusses common existing on demand routing protocols. No existing protocols are fully secure from attacks being encountered in the MANETs. The recent research efforts have made big development on ad hoc network routing, both in theory and in practical implementation. In this paper, Enhanced Efficient Secure Un Observable On Demand Routing Protocol (EESUOR) is proposed based on trust based construction of nodes, group signature and ID-based encryption for ad hoc networks. The design of EESOR offers strong anonymity, complete unlinkability and content unobservabilit for ad hoc networks.

REFERENCES

1. Perkins, C., E. Belding-Royer and S. Das, 2003. Ad hoc on-demand distance vector (aodv) routing, in Internet Engineering Task Force(IETF) draft, July 2003.
2. Moghim, N., F. Hendessi and N. Movehhedinia, 2002. An improvement on ad-hoc wireless network routing based on aodv, in the 8thInternational Conference on Communication Systems ICCS, vol. 2. IEEE, 2002, pp: 1068-1070.
3. Adrian Perrig, Ran Canetti, Dawn Song and J.D. Tygar, 2001. Efficient and Secure Source Authentication for Multicast. In Network and Distributed System Security Symposium, NDSS ’01, pages 35-46, February 2001.
4. AnandPatwardhan, Jim Parker and Anupam Joshi, 2005. Secure Routing and Intrusion Detection in Ad Hoc Networks”.[On-line] accessed on 6th November, 2005 at URL http://csrc.nist.gov/mobilesecurity/ Publications/ nist-umbc-adhocids-ipv6.pdf.
5. Zhu, B., Z. Wan, F. Bao, R.H. Deng and M. KankanHalli, 2004. Anonymous secure routing in mobile ad-hoc networks, in Proc. 2004 IEEE Conference on Local Computer Networks, pp: 102-108.
6. Song, L., L. Korba and G. Yee, 2005. AnonDSR: efficient anonymous dynamic source routing for mobile ad-hoc networks, in Proc. 2005 ACM Workshop on Security of Ad Hoc and Sensor Networks, pp: 33-42.
7. Defrawy, K.E. and G. Tsudik, 2011. ALARM: anonymous location-aided routing in suspicious MANETs, IEEE Trans. Mobile Comput., 10(9): 1345-1358.
8. “Privacy-preserving location-based on-demand routing in MANETs,” IEEE J. Sel. Areas Commun., 29(10): 1926-1934.
9. Kong, J. and X. Hong, XXXX. ANODR: anonymous on demand routing with untraceable routes for mobile ad-hoc networks”, in Proc. ACM MOBIHOC’ 03.

10. Zhiguo, KuiRen and Ming Gu, 2012. USOR: An Unobservable Secure On-DemandRouting Protocol for MANETs, IEEE Transactions on wireless communications, May 2012.
11. Weoliu and Ming Yu, 2014. AASR: Authenticated Anonymous Secure Routing for MANETs in Adversarial Environments, IEEE Transactions on Vehicular Technology Mar 2014.
12. Jeong, I.R., J.O. Kwon and D.H. Lee, 2006. A Diffie-Hellman key exchangeprotocol without random oracles, inProc. CANS 2006, vol. LNCS 4301, pp: 37-54.
13. Dong, D., M. Li, Y. Liu, X.Y. Li and X. Liao, 2011. Topological detection onwormholes in wireless ad hoc and sensor networks, IEEE/ACM Trans. Netw., 19(6): 1787-1796.
14. "Privacy-preserving location-based on-demand routing inMANETs, IEEE J. Sel. Areas Commun., 29(10): 1926-1934.
15. Zhang, Y., W. Liu and W. Lou, 2005. Anonymous communications in mobilead hoc networks," in 2005 IEEE INFOCOM.
16. Zhu, B., Z. Wan, F. Bao, R.H. Deng and M. Kankan Halli, 2004. Anony-mous secure routing in mobile ad-hoc networks, in Proc. 2004 IEEEConference on Local Computer Networks, pp: 102-108.
17. Capkun, S., L. Buttyan and J. Hubaux, 2003. Self-organized public-keymanagement for mobile ad hoc networks,"IEEE Trans. Mobile Comput., 2(1): 52-64.
18. Pfitzmann, A. and M. Hansen, 2000. Anonymity, unobservability andpseudonymity: a consolidated proposal for terminology, draft.
19. Venkatraman, L. and D.P. Agrawal, 2000. A novel authentication in ad hocnetworks, in: Proceedings of the Second IEEE Wireless Com-munications and Networking Conference, Chicago, September 2000.
20. Perkins, C.E. and E.M. Royer, 1997. Ad hoc on demand distance vector(AODV) routing IETF Internet Draft. 1997http://www.ietf.org/inter-net-drafts/draft-ietf-manet-aodv-00.txt.
21. Venkatraman, L. and D.P. Agrawal, 2003. Strategies for enhancing routingsecurity in protocols for mobile ad hoc networks, Journal of Paralleland Distributed Computing, 63(2): 214-227. Special Issue onRouting in Mobile and Wireless ad hoc Networks, Year ofPublication, 2003, ISBN 0743-7315
22. Yi, S., P. Naldurg and R. Kravets, 2002. Security-aware ad hoc routing protocolfor wireless networks, The Sixth World Multi-Conference onSystemics, Cybernetics and Informatics (SCI 2002).
23. Camp, T., J. Boleng and V. Davies, 2002. A survey of mobilitymodels for ad hoc network research, Wireless Communicationsand Mobile Computing, 2: 483-502.
24. Carter, S. and A. Yasinsac, 2002. Secure Position Aided Ad HocRouting, Proceedings of the IASTED International Conferenceon Communications and Computer Networks (CCN02), pp: 329-334, Nov. 4-6, 2002.