NCTS-DCR: Node Centric Trust based Secure Dynamic Source Routing Protocol

Prasuna V G
Associate Professor.
Basaveswara Institute of Information Technology,
Barkathpura, Hyderabad, INDIA,

S Madhusudahan Verma
Professor & Head. Dept of OR&SQC,
Rayalaseema University, Kurnool,
Andhra Pradesh , INDIA

ABSTRACT

An ad hoc network comprises of few particular connections which collectively collaborate to assist other connections to converse with its associates with the assistance of direct wireless broadcasting. Routing issue in ad hoc broadcasting revised routing problem in a unfavorable situations taking into assumption a secure surrounding. A Node Centric Trust based Secure Dynamic Source Routing (NCTS-DSR) standard is recommended which is built on a imprudent line of attack named dynamic source routing (DSR).

Keywords

Manet, Routing protocol, Dynamic source routing, NCTS-DSR, ARIADNE

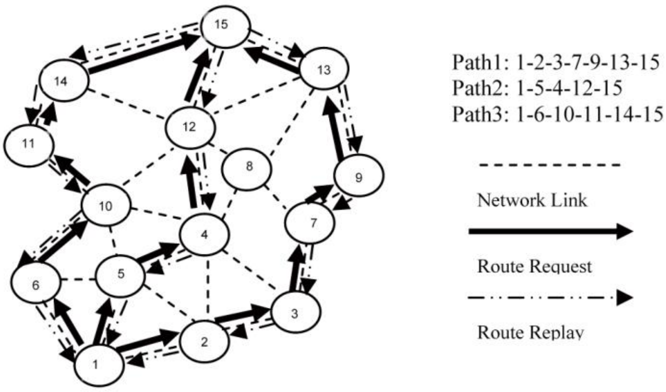
1. INTRODUCTION

Academic and industrial explorations have deduced Wireless ad hoc networks to be an incredibly dynamic arena concerning the predictable extensive claims which have no relevance to any preset communications. The junctions in ad hoc networks are conventionally restricted in terms of their power mines, calculative analysis and transmission networks which is susceptible to confrontations because of various issues like unavailability of communicating modes and difficulty in securing those junctions and so on. Routing standards for MANETS have been initiated and also various methods have been suggested to safeguard the different ad hoc routing standards like SRP, SAODV, Ariadne, ARAN, SADSR, SEAD and SLSP. Futile attempts have been made to safeguard these standards which are heavily susceptible due to erratic nature of the reserved hosts and deficiency in securing hardware, for which Buttya'n and Vajda[4] and Acs et al[5, 6, 7]proposed methods to secure them.

2. RELATED WORK

An on-demand protocol DSR is conceived to confine the bandwidth that is in the width of the control packets by discarding the periodic table update messages which are essential tin proactive routing standards in the absence of a beacon. The routing issue is segregated into two regions namely, route discovery and route maintenance. A favorable route is supposed to be detected for a junction to converse with another junction so as to enable itself to transfer packets to the target node.

A Route Request packet is broadcasted by the originator to ascertain the target for which the route is required. The request is resent to the originator after the junction receives the Route Request, after which a Route Reply is asserted back. A route response is received by the target junction after that the originator will choose a path with the least latency. Every Path Request packet is accompanied by a distinct sequence number and the route that it has travelled from. The series number on the packet has a purpose of avoiding loop configuration and several broadcastings, is cross-checked before it is promoted. A path cache is initiated for the advantage of the transitional junctions to handle the data concerning the paths and originator present in data packet. If a transitional junction that received a Path Request presets a path to the target junction in its path cache, a reply is forwarded to the junction by means of a Path Reply message along with the complete path information. Recurrent Path Request packets submerging the network can be shunned by making use of an exponential back-off algorithm. Piggy-backing of data is also considered to be possible.



If a transitional junction in a route is the cause of a wireless link to break its course in the maintenance method, a Path Error message is produced from the junction which is just next to the broken link so that it convey the new junction of the issue by then which the originator's functional junction recriminates the path establishment process. On receiving a Path Error data packet, the cached records concerning the transitional and the originating junction are discarded. Figure 2 illustrates the working of DSR. If a node 1 is interested to communicate with node 15, a Path Request message is transmitted to all its remaining neighbors, wherein every neighbor ensures its path cache houses the target information. It forwards a Path Reply back to the originator, if it fails to match itself, the Path Request is forwarded to all its neighbors, wherein to dodge loops, Every neighbor cross checks whether it has already sent the Path Request with the respective sequence numbers, after which the target junction responds to all the received Path Request with a message. Information is sent through the track that has the least hop count, 1-5-4-12-15 in the considered example. If there is breakage between the links 12 and 15, the node 12 transmits a Path Error message to the originator, which will retransmit the Path Request message. The tale entry from their path cache will be discarded and no defense issues will be produced in the basic DSR arrangement as also the resource supervision is exploited.

QoS Guided Route Discovery: QoS Guided Route Discovery standard was organized by Maltz[13] which permits a junction to identify QoS metrics which needs to be originated to forward a Path Request, which when commenced searches for it in its cache path. QoS Guided Route Discovery is executed in bid to ascertain a more convincing path only if there is a successful provision of flow formation. A second search is advised wit the pretention of a bit higher rate of resources which is accessible along with the already existing path. The QoS Guided Route Discovery procedure assists a junction in transmitting a Path Request in the optional QoS Request Header which designates the resource type, minimum acceptable resource level and the present path resource level. There may be presence of one or more QoS Request Headers in a Path Request message which is received by a junction where every QoS Request Header deals with other headers to ensure whether the junction is able to prop up new resources flow at a level which is at the minimum equivalence to the requested one. The Path Request message is rejected and in lieu of it unable to brace the present level mentioned in the QoS Request Header, the Path Request is forwarded as if done under normal circumstances. The three conventional QoS metrics are being specified and used by Maltz which are bandwidth, latency and jitter. A junction that sends the data is capable of revising the new update level that contains a value which is less than the resource. Latency and jitter ensure that their values in every junction is mentioned in the Request and summed up with the obtained value. Appropriate paths throughout the network can be discovered via a routing standard that makes use of QoS Guided Route Discovery which must confine those flow resources on a best attempt basis. A Route institution and a resource corollary standard is essential to be utilized wherein an initiator launches a packet flow by forwarding Establish Flow packet traversing the route, where every node n the path preserves the resources exclusively for the flow an further forwards the packet to the next junction. A Flow Error packet is sent to the flow originator when the junction fails to meet the QoS requirements. Two extra packet types are entailed for stream enterprise; they can be validated via transmission verification or via usage of pair-wise validation making use of shared keys. When a junction validates the originator, policies are employed to ascertain whether or not the originator is validated to preserve these resources. Flow Error packets as well as Path Error packets can be validated by on-demand routing standards.

Observation: An alarming issue in Qos Guided route discovery standard is specifying that resources are accessible at any specific junction and also that a junction should avoid a REQUEST if it houses a much better REQUEST which brings out three problems mainly, a transitional junction may be unaware of which trade-offs of various QoS metrics are favored by the initiator. Next, an assaulter is capable of forcing a junction to send many Path Requests by transmitting a single Request many times, making use of enhanced metrics. Thirdly, if there is a provision of a junction sending a REQUEST, many sent packets arise from a single direction discovery request.

Securing Quality of Service Route Discovery: QoS Guided Route Discovery gave wings for SQoS[15] to evolve, which is a superior provision for on demand ad hoc network routing. This form completely depends on symmetric cryptography whose primal are in three to four orders of magnitude that is far more quick and efficient as compared with asymmetric cryptography.

Hash and M chains are the basis for building SQoS. Hash function is considered to be a one way function which can be explained as follows. Y=H(X), where X can be any arbitrary value and H is considered to be the hash function. Hash chain is fulfilled by applying the hash function continuously .If at all we have X, then there is every possibility of getting H(X), H(H(X)), H(H(H(X)))… HN(X), by applying hash function N times, where N indicates specified parameter. The last calculated value should be known to the receiver. For the purpose of authentication, the sender must and should send the (N–1)th value of the hash chain to the destination. Whenever the receiver receives it, it makes the hash function to apply one time. In the case of the result matching the stored value, the sender is authenticated by it. After that the sender has to send (N–2)th value of the hash function for the purpose of next authentication. As the receiver already has the (N-1)th value, the receiver once again applies the hash function to the value received for comparing it with (N–1)th value. It means that, every user password is the value that is required by the system the system for authenticating the next password.

MW chain gives on the spot authentication and low storage overhead. MW chain depends on one time signature. The working of one time signature works is that each and every node chooses a private key K which is utilized for producing verification key V and signature S. If the case of node having a message to send, it will sign it utilizing its signature S. The nodes which have been communicated key V only will be able to read the message (Observe that node that has V, will not be able to generate S). By this way we will be able to sign each message with distinct S (derived from K), and check it utilizing either distinct V or in the same V. MW chain consists of the same properties of a hash chain and also has the additional property that a signature S utilizing key. Ki+1 can be utilized for producing key Ki utilizing the equation Ki = f(s, m).However, it cannot be utilized to drive Ki+1. It means that, whenever, a node A has a private key Ki+1, then S = G (Ki+1), and V = G(S) and G must and should be a secure one-way hash function. Going by example, suppose node A is sending a message m to node B. The message that is sent by node A is m signed with S in the format of f(s, m). B can get Ki from knowing f(s, m). Now B has the new private Key Ki, that can be utilized later in the process of communicating A.

Observation: At SQoS, the hash chain has been usurped with an MW chain for preventing the modification of the unchangeable fields of the request. A node utilizes one MW-chain step for each route discovery, and utilizes the signature S from that MW-chain step for authenticating the unchangeable fields of Route Request message. SQoS solves the three problems which have been earlier recognized for the simple QoS scheme by giving the source with control over route Request message that are being re-forwarded. In SQoS the initiator gives a specific list of metrics of interests like latency and bandwidth. For each and every metric, the initiator specifies the levels that are maximum necessary and minimum desirable , the length of the hash chain and by which way the steps are to be divided, linearly or logarithmically. The focus of SQoS on secure quality of service gives guaranty such as bandwidth and latency by utilizing MW chain. But, SQoS didn’t do the discussion of whether the intermediate nodes would be able to whatever it obeys to support. Significant factors such as the node power, CPU, RAM, encryption capability, exposure to other nodes and the organizational hierarchy have not been dealt in the route computation process.

Ariadne — Ariadne [16] is a secure on-demand routing protocol that are used in ad hoc networks. The authentication of data in this protocol is given by 3 different techniques: digital signature, MACs, or TESLA [9]. Here the exploration of Ariadne was carried out with MACs. The design of Ariadne is depended on DSR. In the same way DSR has, it comprises two basic phases, route discovery and route maintenance [8]. The route discovery comprises 2 stages. (l) Route request stage-the network is flooded with a route request packet (RREQ) by the source node, and each node (with the exception of destination) rebroadcasts it. (2) Route reply stage-On receiving a RREQ the destination sends a route reply packet (RREP) in path that is reverse of the RREQ.

The source node produces a RREQ and sends it to its neighbors. The RREQ has the identifiers of the source (no) and the destination (n)), a haphazard request identifier (id), and a MAC (mno) computed over these elements with a key that is shared together by the source and the destination (KnOn))'. The hashing of this MAC iteratively is done by each intermediate node together with its own identifier utilizing a publicly known one way hash function (h). The hash values calculated by this way are known as per-hop hash values. Each and every intermediate node that gets the RREQ for the first time re-calculates the per-hop hash value and then add its identifier to the list of identifiers in the RREQ, and calculates a MAC (mnj) on the RREQ, that is updated, with a key that it shares with the destination. Lastly, the MAC is attached to a MAC list in the RREQ, and the RREQ is re-broadcast.

Whenever the destination gtes the RREQ, it checks the per-hop hash by re-computing the source' s MAC and the per-hop hash value of each and every intermediate node. After that it checks all the MACs in the RREQ. In the case of all these verifications being successful, then the destination produces a RREP and sends it back to the source through the reverse of the route that is obtained from the RREQ. The RREP has the identifiers that belongs to the source and the destination, the route obtained from the RREQ and the MAC of the destination (mno.nl) on all these elements with a key that is shared by destination and the source. Each and every intermediate node sends the RREP to the next node on the route (towards the source) without any changes. Whenever the source gets the RREP, it checks the MAC of the destination. In the case of the verification being successful, then it obeys the route that is sent back in the RREP [10].

The authors of [11] choosed low-overhead version, as it is more effective than the basic protocol in tenns of computational and communication overhead. A mono MCA restored with the intermediate nodes repeatedly in the route despite a per-hop hash value or MCA group.The route request re broadcast by the i-th intermediate node n_i has the following form:

$msg_{rreq} = \{ rreq, n_0, n_l, id, (n_1, \ldots., n_i), m_{ni} \}$

where m_{ni} is a MAC countted by n_i with k_{ninl} on the route request that get from n_{i-1}

$msg_{rreq} = \{ rreq, n_0, n_l, id, (n_1, \ldots., n_{i-1}), m_{n-1}\}$

However the RREP of this version is as equal as the RREP in the first version.

Observation: Ariadne is weak where an attacker can easily identify the route. The feedback is not possible in this version because which node send packets is not sure (even that DSR itself is based on past history through including the full route through the route request, hop-by-hop) [2].

We can find secret channels in this Ariadne like node identifiers, id, the MACs in the routing messages. The attackers utilize the secret channels to communicate and share useful information in the group, because intermediate nodes won't test the secret channels performance [11]. The clever opponent will be seen as -y-x [10] which restrains x opponent nodes and utilizes y adjust identity. In reference [7], Acs et al.[5, 6, 7] proposed an Active-I-2 attack on Ariadne-MAC and an Active-2-2 attack on low-overhead version of Ariadne-MAC. Moreover, Ariadne struggles with a dispute that in the route answer period every intermediate node cannot check RREPs. Mono Active-I-I opponent will manufacture a RREP, the additional valid intermediate nodes will not find the malfunction yet send manufacture RREP still it reaches base. Though the fabricated RREP seeks by the base it will not dislodge (or locate) the harmful node. Such difficulties are the main cause which lessen the ability of protocol in dislodging the harmful nodes . In Figure 2 (b) an Active-I-I display offenses on this protocol.

CONFIDANT — CONFIDANT [17] is a safe of routing protocol in avoiding the malfunction of nodes unimportant to additional nodes to communicate with. It is depends on selective altruism and utilitarianism. It's main objective is find and eliminate malfunctioning of nodes, by decreasing its influence . Belief and routing conclusions will be depend on experienced, identified and sending information to additional nodes. The design of CONFIDANT believes that the network layer depends on DSR. CONFIDANT includes other components like the monitor, the reputation system, the path manager, and the trust manager. Each component performs its function with its name. where monitor is for the neighborhood nodes to record communication between additional nodes. The trust manager handles incoming and outgoing ALARM messages. ALARM information will be forwarded by the trust manager of a node to send notice to additional harmful nodes. The reputation system helps in avoiding a centralized rating, local rating lists and/or black lists carry on by the every node and potentially exchanged with friends. Likewise reputation systems help in some online auctioning systems. They supply a means of getting a quality rating of participants of transactions by having both the buyer and the seller give each other feedback on how their activities were viewed and calculated. Path manager handles the following functions like path re-ranking according to security metric, e.g. reputation of the nodes in the path, deletion of paths containing malicious nodes, action on receiving a request for a route from a harmful node, e.g. neglect, with out answering, and action on receiving request for a route consists harmful node in the base route, e.g. neglecting, alert the base.

If the monitor finds an anomaly, it tell the reputation system to perform an action, that controls a local ratings list. These lists are shared with other nodes; the trust monitor deals input with additional nodes. If a list got from a believable node, the receiver will send message to its local ratings list. Unlikely when a list got from unidentified base , the receiver neglects and calculate less importance from list of believed node. Subsequently, the path manager selects paths from the node's route depend on a blacklist and the local ratings list. The path manager responds to REQUEST from a node on the blacklist or to a REQUEST that bisected a node on the blacklist.

Observation: CONFIDANT includes global reputation values. Every node consist a mono recognized value for additional node that it meets, where this value integrates different recognized values. Utilizing global reputations helps in different issues [18]. Exclusively, a global reputation value creates a node to conceal malfunctioning from supporting another function. Global reputation values, however will not say the essentials located on various services by various nodes. The division quality of the mechanism leads to several disturbances in the reputation value.

It will also create invasion on the reputation value like advertising wrong high rating or wrong low rating regarding other node and fictitious variation

3. NODE CENTRIC TRUST BASED SECURE ROUTING [NCTSR] PROTOCOL

3.1 ROUTE DISCOVERY PROCESS

Objective of the NCTS-DSR route establishment process is preventing unauthorized hops to join in root during route request process

Privileges assumed at each node that exists in the network:

- The node that belongs to the network contains the capabilities fallowing
- Able to generate hash method based id for broadcasting packets
- Ability to issue digital certificate
- Ability to maintain the id of hop from which egress data received and id of hop to which ingress data
- Elliptic Curve based cryptography functionality will be used to protect data transmission

Hop node registration process

Hop nodes exchange their digital certificates recursively with a time interval ς . The delay between two iterations represented by an interval referred as certificate exchange interval ς. Each node submits its certificate to one and two hop level nodes.

$$\varsigma_h = t/dt_t$$

ς_h is time interval for node h to submit its digital certificate to neighbor hop nodes

' ι ' is interval threshold

' dt_ι ' is distance that can travel by a node h in interval threshold ' ι '

Description of the notations used in route detection process:

n_s	source node
n_d	destination node
n_r	relay node
n_e	node from which egress data received by n_r
$n_{e'}$	node from which egress data received by '$n_{e'}$, and two level hop to n_r
n_i	node to which ingress data send by 'n_r '
Cer_h	digital certificate of hop node h
$addn_h$	address of hop node h
$It(Cer_h) =$ $c\mathrm{Ts}_{nr}$ - iTs_{ne}	$It(Cer_h)$ is certificate life time of the hop node h
CTs_{nr}	Current time stamp at the relay node n_r
iTs_{ne}	Timestamp when cer_{nr} created (cer_{n_r} creation time)
p_{id}	Is RREQ unique ID that generated in secure random way
sp_{id}	is set of packet *ids* those transmitted by a node

3.1.1 Root request process

RREQ packet at source node n_s contains

$< \mathrm{addn}_{s}, \mathrm{addn}_{d}, \; \mathrm{p}_{id}, \mathrm{cern}_{e}, \varsigma_{e}, \mathrm{cern}_{s,}, \varsigma_{ns}, \mathrm{cern}_{i}, \mathrm{ECPK}_{ns} >$

Note: Here $\mathrm{cern}_{e,}$ is null.

Process of RREQ packet validation and construction at first hop node of the source node

Table 1: Algorithm for RREQ packet evaluation at hop node of the source node

```
1. Step 1:
   a. If $p_{id} \in Sp_{id}$ then RREQ packet will discarded
      else adds $p_{id}$ to $Sp_{id}$ and continues step 2
2. Step 2:
      a. If $lt(cern_i)$, is valid and $cer_{ni} == cer_{nr}$ then
      continues step 3, else discards RREQ packet.
      [Here $cer_{ni}$ is $cer_{nr}$ because the current node
      certificate is available at sender node as certificate
      of one hop node that acts as target for ingress
      transaction]
3. Step 3:
      a. If $cern_e$ is null then assumes sender is source
      and continues step 4.
      Else

      b. If $cern_e$ is not null and $lt(cer_{ne}) < \varsigma_{ne}$

      and $cern_e = cern_{e'}$ then is $cern_e$ valid and
      continues step 4, else RREQ will be discarded. is
      $cern_{e'}$ certificate of the node that exists as two hop
      level to current rely node.

      [Here for $cern_e$ senders node is $cern_{e'}$ for current
      rely node]
     $lt(Cer_{ne}) = cTs_{nr} - iTs_{ne'}$
      Here:
      $iTs_{ne}$ is timestamp at current relay node
```

```
        $cern_e$ is certificate carried by RREQ packet
  4. Step 4:
        a. $lt(cer_{ne}) < \varsigma_{ne}$ and $cern_e = cern_{e'}$ then source
           node $n_s$ is valid and continues step 5

        b.If $cern_e$ is valid then that RREQ packet will be
          considered and continues step 5 else that packet
          will be discarded.

  5. Step 5:

        a. If $addn_d \neq addn_r$ then Update the RREQ
           packet $< addn_s, addn_d, addn_r, p_{id}, cern_e, \varsigma_e,$
           $cern_r,, \varsigma_{nr}, cern_i, {}_{ECPKns} >$ and transmits to $n_i$

        b. Else if $addn_d = addn_r$ , nr is identified as
           destination node and starts RREP process
```

```
  1. Step 1:
     a. If $p_{id} \in Sp_{id}$ then RREQ packet will discarded
        else adds $p_{id}$ to $Sp_{id}$ and continues step 2
  2. Step 2:
     a. If $lt(cern_i)$ is valid and $cer_{ni} == cer_{nr}$ then
        continues step 3
        Else discards RREQ packet.
     [Here $cer_{ni}$ is $cer_{nr}$ because the current node
      certificate is available at sender node as certificate
      of one hop node that acts as target for ingress
      transaction]
  3. Step 3:
        a. If $cern_e$ is not null and $lt(cer_{ne}) < \varsigma_{ne}$
           and $cern_e = cern_{e'}$ then is $cern_e$ valid
           and continues step 4, else RREQ will be
           discarded.
        b. $cern_{e'}$ is certificate of the node that exists
           as two hop levels to current rely node.
           [Here $cern_e$ for senders node is $cern_{e'}$ for
           current rely node]

           $lt(Cer_{ne}) = cTs_{nr} - iTs_{ne'}$

           Here:
           $cTs_{nr}$ is timestamp at current relay node
           $Cer_{ne}$ is certificate carried by RREQ packet
  4. Step 4:
        .a. $lt(cer_{nr}) < \varsigma_{ne}$ and $cern_r = cern_e$
        then source node $n_r$ is valid and continues to
        step 5 else discards the RREQ packet
  5. Step 5:
   . If $addn_d \neq addn_r$ then Update the RREQ packet
        $< addn_s, addn_{d,}, p_{id}, ( addn_1, addn_{2,} \ldots\ldots\ldots addn_{r-2}, addn_{r-1} addn_r ), cern_{r,}, \varsigma_{nr}, cern_i, {}_{ECPKns} >$
        and transmits to $n_i$

        b. Else if $addn_d = addn_r$ , $n_r$ is identified as
        destination node and starts RREP process
```

Process of RREQ construction at relay hop node of the source node

Once packet received by next hop (in that packet referred as n_i) then continues the above four steps in sequence with minor changes, described here:

Table 2: Algorithm for RREQ packet evaluation at relay node that is not hop node to source node

When compared algorithm in table 2 with algorithm in table 1, a change can be observable at step 3, we are not accepting certificate cer_{n_e} carried by RREQ as null, since n_r representing in RREQ packet is not source node.

3.1.2 RREP process

Once n_d receives RREQ it performs verification as mentioned in table 2. Upon successful validation,
It performs fallowing functionality.
If RREQ that was received is valid then

It collects $ECPK_S$ and calculates $ECPK_d$ (Elliptic curve cryptography approach explained in next section B).
Then it constructs RREP packet at n_d as follows:
$< addn_s, addn_{d,} {}_{ECPKnd}, lst_{nr,} , cern_{e,} , cern_i , cern_{d,} \varsigma_{nd} \; p_{id} >$

Since the RREP packet constructed at n_d cer_{ne} is null.

Process of RREP packet validation and construction at first hop node of the destination node

Table 3: Algorithm for RREP packet evaluation at hop node of the destination node

```
➔Here $n_r$ is hop node of the $n_d$
   1. Step 1:
      a. If $n_r \in list_{nr}$ then continues step 2 else discards
         RREP packet
      b. If $p_{id} \in Sp_{id}$ then RREP packet will discarded
         else adds $p_{id}$ to $Sp_{id}$ and continues to step 2
   2 . Step 2:
         If $lt(cern_i)$ is valid and $cer_{ni} == cer_{nr}$ then continues
         step 3, else discards RREP packet.
         [ Here $cer_{ni}$ is $cer_{nr}$ because the current node
         certificate is available at sender node as certificate of
         one hop node that acts as target for ingress
         transaction]
   3 Step 3 :
       a. If $cer_{n_e}$ is null then assumes sender is source of the
          RREP and continues to step 4.
       b. Else If If $cern_e$ is not null and
          $lt(ce\ r_{ne}) < \varsigma_{ne}$ and $cern_e = cern_{e'}$ then is $cern_e$
          valid and continues step 4, else RREP will be
          discarded.
       $cer_{n_e'}$ is certificate of the node that exists as two hop
          level to current rely node.
[Here $cer_{ne}$ for sender’s node is $cer_{ne'}$ for current rely node
$lt(Cer_{ne}) = cTs_{nr} - iTs_{ne'}$
$cTs_{nr}$ is timestamp at current relay node $Cer_{ne}$ is certificate
carried by RREP packet ]
```

```
4. Step 4: . $lt(cer_{nd}) < \varsigma_{nd}$ and $cern_d = cern_e$
       then source node $n_r$ is valid and continues to step
       5. Else that packet will be discarded.
5. Step 5:
       a. If $addn_s \neq addn_r$ then Update the RREP
     packet as
 $< addn_s , addn_{d,}\ _{ECPKnd} , lst_{nr,} , cern_{e,} , cern_r , cern_{i,} \varsigma_{nr}\ p_{id} >$
   and transmits to $n_i$ .
     b. Else if if $addn_s = addn_r$ $n_r$ identified as source node
     and stops RREP process
```

Process of RREQ construction at relay hop node of the source node

Once RREP packet received by next hop (in that packet referred as n_i) then verifies and continues the process as described in table 4:

Table 4: Algorithm for RREP packet evaluation at relay node that is not hop node to destination node

```
1. Step 1: If $n_r \in list_{nr}$ then continues step 2 else discards
          RREP packet
2. Step 2 : If $p_{id} \in Sp_{id}$ then RREP packet will discarded
           else adds $p_{id}$ to $Sp_{id}$ and continues to step 3
3. Step 3: If $lt(cern_i)$ is valid and $cer_{ni} == cer_{nr}$ then
          continues step 4, else discards RREP packet.
   [ Here $cer_{ni}$ is $cer_{nr}$ because the current node certificate is
     available at sender node as certificate of one hop node that
     acts as target for ingress transaction]

4. Step 4: If $cer_{n_e}$ is null and
          $lt(ce\ r_{ne}) < \varsigma_{ne}$ and $cern_e = cern_{e'}$ then is $cern_e$
          valid and continues step 5, Else RREP will be
          discarded.
        $cer_{n_{e'}}$ is certificate of the node that exists as two hop
          level to current rely node.
[Here $cern_e$ is for senders node is $cern_{e'}$ for current rely node.
$lt(Cer_{ne}) = cTs_{nr} - iTs_{ne'}$
     $cTs_{nr}$ is timestamp at current relay node $Cer_{ne}$ is certificate
     carried by RREP packet ]
5. Step 5: . $lt(cer_{nr}) < \varsigma_{ne}$ and $cern_r = cern_e$
        then node $n_r$ is valid and continues to step 6. Else
        the RREP packet
6. Step 6:
   a. If $addns \neq addn_r$ then Update the RREP packet as<
$addn_s , addn_{d,}\ _{ECPKnd} , lst_{nr,} , cern_{e,} , cern_r , cern_{i,} \varsigma_{nr}\ p_{id} >$
   and transmits to $n_i$ .
     b. Else if if $addn_s = addn_r$ $n_r$ identified as source node
     and stops RREP process
```

3.2. ELLIPTIC CURVE CRYPTOGRAPHY FOR CONSTRAINED ENVIRONMENTS

We have to determine the “hard problem” that is related to multiplication of couple of primes or considering the separate distinct algorithm otherwise named as discrete algorithm, in order to make a cryptographic system utilizing elliptic curves.

Let us assume a function Q=kP, her Q and P are subset of elliptic curve on GF(2^n) and. K < 2_n.Comparatively we can simply determine the value of Q given k and P, than that of when K given Q anD P. This is termed as the discrete algorithm problem for elliptic curves.

3.2. 1. Key Exchange

The subsequent procedure is a way in which key exchange shall be achieved. Selecting q = 2_n as a huge integer parameters of elliptic are a and b. By the previous selections a set of points are described. Then we have to select a point called base point that has huge value represented by n. The attributes of Elliptic E and G will be intimated to the contestant.

For an illustration let us considers couple of consumers naming A and B and the key exchange between the two shall be consummated given below:

1. A value n_A will be chosen by A that is lower than that of n which is considered as a confidential key for consumer A. Later consumer A produces a public key $ECPK_A = n_A$*G subsequently public key $ECPK_A$ is assigned on to E.

2. Consumer B correspondingly chooses a confidential key n_B by which calculates the public key $ECPK_B$

3. Consumer A produces a clandestine key
$K = n_A *ECPK_B$and consumer B produces the clandestine key.
$K = n_B *ECPK_A$
The computations that are present in 3rd statement derive an equal output.

$$n_A *ECPK_B \equiv n_A *(n\equiv n_B *(n_A *G) \equiv n_B * ECPK_{A\,B} *G)$$

Strong point present in the key exchange procedure is that it is not easy to dissolve this format. That is when an cracker tries to hack the procedure then had to calculate the value of K given G and kG. But this will be durable and most of the times it is unlikely to occur when it comes to a constrained situations.

4.SIMULATION AND RESULTS DISCUSSION

The tool that was utilized in accomplishing the test was NS 2. Considering the mobility and amount ranging from 20 to 200, a simulation network simulation network has been constructed. The attributes and the values of the simulation are explained in the below table 5. If the packet that was sent is legal then it confirms that buffer is assigned successfully. The main goal of this model is to contrast the Ariadne [16] and NCTS-DSR. In order to check the better of the two, a test is conducted by using some of the severe problems that are motioned underneath and the corresponding outputs are given in the table 6, which show the capability of the procedures in facing these problems.

- Rushing attack
- Denial of service
- Routing table adaptation
- Tunneling

The pros of the NCTS-DSR than that of the Ariadne is the procedure of guarding the tunneling attack.

Table5: Simulation attributes taken for the test.

Number of nodes Range	50 to 200
Dimensions of space	1500 m × 300 m
Nominal radio range	250 m
Source–destination pairs	20
Source data pattern (each)	4 packets/second
Application data payload size	512 bytes/packet
Total application data load range	128 to 512 kbps
Raw physical link bandwidth	2 Mbps
Initial ROUTE REQUEST timeout	2 seconds
Maximum ROUTE REQUEST timeout	40 seconds
Cache size	32 routes
Cache replacement policy	FIFO
Hash length	80 bits
certificate life time	2 sec

Table 6: The capabilities of various routing protocols to defend from several problems.

Proposed protocols	Routing strategy	Protects from Rushing attack	Protects from Denial of service	Protects from Routing table modification	Protects from Tunneling
ARAN	On demand	Yes	No	Yes	No
SAR	On demand	Yes	No	Yes	No
SRP	On demand	Yes	Yes	Yes	No
SEAD	Table driven	Yes	Yes	Yes	No
Ariadne	**On demand**	**Yes**	**Yes**	**Yes**	**No**
SLSP	Table driven	Yes	Yes	Yes	No
SAODV	On demand	Yes	No	Yes	No
CORE	Table driven	No	yes	No	No
CONFIDANT	On demand	Yes	No	No	Yes
BYZANTINE	On demand	Yes	Yes	Yes	No
WATCHDOG & PATHRATER	On demand	No	No	No	Yes
NCTS-DSR	**On demand**	**Yes**	**Yes**	**Yes**	**Yes**

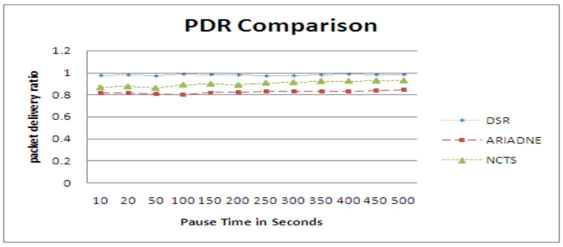
Figure 2(a) illustrates Packet Delivery Ratio (PDR) for DSR, ARIADNE and NCST. By considering this output it is enough to prove that NCST manages maximum failure of PDR than that of ARIADNE in case of DSR. Fairly accurate failure amount of PDR that is restored by the NCST than ARIADNE is 1.5%. This is balanced amount among the pauses. The least amount of restoring examined is 0.18% and the highest id 2.5%. The next Figure 2(b) specifies ARIADNE benefit than that of NCTS in case of Path optimality. NCTS utilized nearly 0.019 hops more when compared to ARIADNE as the reason of hop level certification confirmation method of the NCST which removes the points that are invalid. This is a minor benefit of ARIADNE over NCST that can be examined.The derivation for the packet delivery fraction (PDF) is:

$$P' = \sum_{f=1}^{e} \frac{R_f}{N_f}$$

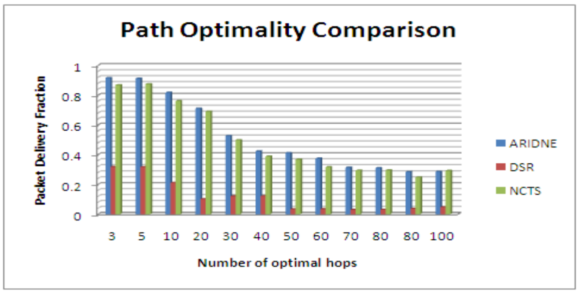
$$P = \frac{1}{c} * P'$$

- P is the division of effectively reached packets,
- C is the overall amount of flow or associations,
- f is the distinctive flow id allocated as index,
- R_f is the amount of packets acknowledged from flow f
- N_f Is the amount of packets transferred to flow f.

Figure 2(c) proves that NCTS is has less packets than that of ARIADNE. This benefit of the NCTS could be feasible as a reason of availability of constant paths without negotiation or offended nodes. The Packet overhead derived in ARIADNE is nearly 5.29% larger than packet overhead derived in NTCS. The least and highest packet overhead in ARIADNE than NCTS derived is 3.61% and 7.29% correspondingly.

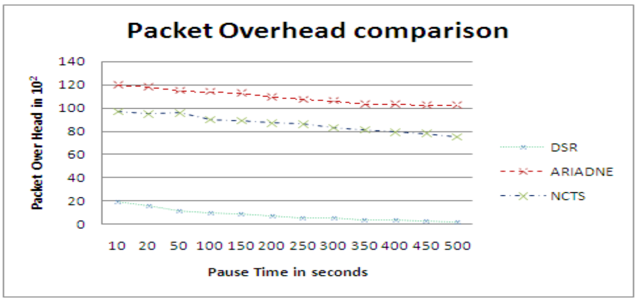


2(a) Packet delivery ratio assessment utilizing line chart

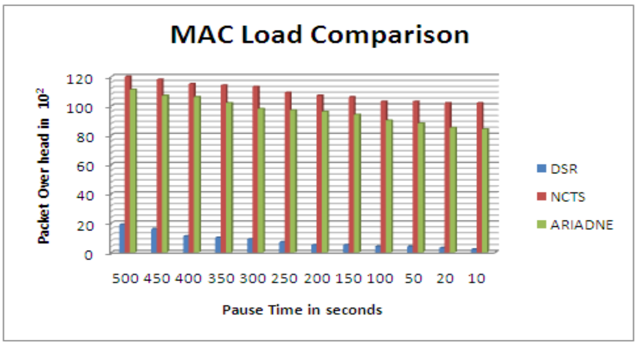


2(b) Bar chart illustration for Path optimality

MAC load overhead is high in NCTS than ARIADNE to some extent. This is viewed in figure 2(d). This is occurred due to the control packet swap in NCTS for adjacent hop verification by utilizing certificate exchange. The common MAC load overhead in NCTS than ARIADNE 1.64%. The least and highest MAC load overhead derived is 0.81 and 3.24% correspondingly. Appealing outputs have been determined for DSR when all the assessment procedures are considered. Apart from path optimality DSR executed fine as a result of not taking security concern into account as a routing attribute, and it is producing enhanced QOS without risk in routing hypothesis. But factually it is false in actual. In path optimality verification DSR place at end as a reason of not taking security restraints into account, amongst three measured procedures, eventually this made to recognize uneven paths.



2(c) A line chart illustration for Packet overhead assessment details



2(d) Mac load assessment illustrated in bar chart format

5. CONCLUSION

Considering DSR approach this paper assessed the security protocols like QoS-Guided Route Discovery[13], sQos[15], Ariadne [16] and CONFIDANT [17]. It illustrated the boundaries and assaults of them which are sensitive and complicated to be generated through familiar logic regarding their characteristics. The projected NCTS-DSR protocol utilizes digital signature exchange on RREQ and RREP. By this they put in the adjacent ones in 2 hops from the node in calculating and deriving them. Malevolent nodes are evaded by these digital signatures that facilitate the protocol, so that they don't involve in involving into the routing and path discovery. This also facilitates in identifying the fallacious routing content and the corresponding nodes.

6. . REFERENCES

[1] ling Liu, Fei Fu, lunmo Xiao, and Yang Lu "Secure Routing for Mobile Ad Hoc Networks", Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and ParallellDistributed Computing, pp. 31 4-31 8. 2007.

[2] Loay Abusalah, Ashfaq Khokhar, and Mohsen Guizani "A Survey of Secure Mobile Ad Hoc Routing Protocols", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, vol. 1 0, no. 4, pp. 78-93, 2009.

[3] Wenchao Huang, Yan Xiong, and Depin Chen , "DAAODV: A Secure Ad-hoc Routing Protocol based on Direct Anonymous Attestation", International Conference on Computational Science and Engineering, pp. 809-81 6, 2009.

[4] L. Buttya'n and I. Vajda, "Towards Provable Security for Ad Hoc Routing Protocols," Proc. ACM Workshop Ad Hoc and Sensor Networks (SASN '04), 2004.

[5] G. Acs, L. Buttya'n, and I. Vajda, "Provably Secure On-Demand Source Routing in Mobile Ad Hoc Networks," Technical Report 1 59, Int'l Assoc. for Cryptologic Research, 2004.

[6] G. Acs, L. Buttya'n, and I. Vajda, "Provable Security of On-Demand Distance Vector Routing in Wireless Ad Hoc Networks," Proc. European Workshop Security and Privacy in Ad Hoc and Sensor Networks (ESAS '05), pp. 1 1 3-1 27, 2005.

[7] G. Acs, L. Buttya'n, and I. Vajda, "Provably Secure On-Demand Source Routing in Mobile Ad Hoc Networks," IEEE Trans. Mobile Computing, vol. 5, no. II, pp. 1 533-1 546, Nov. 2006.

[8] Chu-Hsing Lin, Wei-Shen Lai, Yen-Lin Huang, and Mei-Chun Chou "Secure Routing Protocol with Malicious Nodes Detection for Ad hoc Networks", 22nd International Conference on Advanced Information Networking and Applications, pp. 1 272-1 277, 2008.

[9] X.SU, and R. V. Boppana, " Cross check mechannism to identify malicious nodes" , Security communication network, vol. 2, pp. 45- 54, 2009.

[10] M.Burmester, and Breno de Medeiros, "On the Security of Route Discovery in MANETs", IEEE Trans. Mobile Computing, vol. 8, no. 9, september 2009.

[II] Hu Y-C, Perrig A, Johnson DB, "Ariadne: a secure on-demand routing protocol for ad hoc networks", Wireless Networks 2005; 11(1- 2): 21- 38.

[12] D. B. Johnson, "Routing in Ad Hoc Networks of Mobile Hosts," Proc. IEEE Wksp. Mobile Computing Systems and Applications, Dec. 1994.

[13] W. Liu and Y. Fang, "SPREAD: Enhancing Data Confidentiality in Mobile Ad Hoc Networks," Proc. IEEE INFOCOM 2004, 2004.

[14] D. A. Maltz, "Resource Management in Multi-hop Ad Hoc Networks," CMU School of Computer Science Technical Report CMU-CS-00-150, Nov. 21, 1999.

[15] R. Hauser, T., Przygienda, and G. Tsudik, "Lowering Security Overhead in Link State Routing," Computer Networks, vol. 31, no. 8, Apr. 1999, pp. :885–94.

[16] Y. Hu, A. Perrig, and D. B. Johnson, "Packet Leashes: A defense against Wormhole Attacks in Wireless Ad Hoc Networks," Proc. IEEE INFOCOM 2003, vol. 3, Apr. 2003, pp. 1976–86.

[17] S. Buchegger and J. L. Boudec, "Performance Analysis of the CONFIDANT Protocol Cooperation Of Nodes-Fairness In Dynamic Ad-hoc Networks," Proc. IEEE/ACM Symp. Mobile Ad Hoc Networking and Computing (MobiHOC), 2002.