A Scalable and Secure Position-Based Routing Protocol for Ad-Hoc Networks

Liana Khamis Qabajeh[1] Miss Laiha Mat Kiah[1] Mohammad Moustafa Qabajeh[2]
liana_tamimi@ppu.edu misslaiha@um.edu.my m_qabajeh@yahoo.com
[1]*Faculty of Computer Science and Information Technology, University of Malaya, Malaysia*
[2]*Department of Electrical and Computer Engineering, IIUM, Malaysia*

ABSTRACT

Mobile Ad-Hoc Networks (MANETs) are becoming increasingly popular as more and more mobile devices find their way to the public. A crucial problem in Ad-Hoc networks is finding an efficient route between a source and a destination. Due to MANET's inherent characteristics, secure routing may be one of the most difficult areas to tackle because opponents can add themselves to a MANET using the existing common routing protocols. Hence, this paper proposed a new model of routing protocol called ARANz, which is an extension of the original Authenticated Routing for Ad-Hoc Networks (ARAN). Apart from the authentication methods adopted from ARAN, ARANz aims to increase security, achieve robustness and solve the single point of failure and attack problems by introducing multiple Local Certificate Authority servers. Moreover, by dealing with the network as zones and using restricted directional flooding, our new model will exhibit better scalability and performance. An overview and a qualitative comparison between ARANz and some existing Ad-Hoc routing protocols is presented in this paper.

Keywords*: position-based routing, secure routing, scalable routing, ad-hoc networks, wireless networks.*

1.0 INTRODUCTION

Ad-Hoc wireless networks are self-organizing multi-hop wireless networks, where all the hosts (or nodes) take part in the process of forwarding packets. Ad-Hoc networks can quickly and inexpensively be set up as needed since they do not require any fixed infrastructure, such as base stations or routers. Therefore, they are highly applicable in many fields, such as emergency deployments and community networking.

A key component of Ad-Hoc wireless network is an efficient routing protocol since all the nodes in the network act as routers. Ad-Hoc network routing protocols are difficult to design in general. There are two main reasons for that; the highly dynamic nature of the Ad-Hoc networks due to high mobility of the nodes, and the need to operate efficiently with limited resources such as network bandwidth, CPU processing capacity, memory and battery power of each individual node in the network. Moreover, the concept and structure of Ad-Hoc networks make them prone to easy attack through several ways such as modification, impersonation, and fabrication.

Considering the Ad-Hoc networks environments, the managed-open environment is the one that we are most likely to see expanding in the nearest future. Such an Ad-Hoc network might be formed by peers at a conference, or students on a campus. In this type of environment, the possibility to use already established infrastructure to some extent to help us secure the Ad-Hoc network is available. This means that there is an opportunity for pre-deployment or exchange of public keys, session keys, or certificates. This opens up a whole new range of strategies that use certificate servers and other similar software to provide a starting point for the security in the network.

For example, without online trusted servers as in wired networks, it is difficult to be acquainted with the trustworthiness of each node, thus keeping away malicious nodes from the routes. However, the approach where one centralized server is used in the Ad-Hoc network is not practical as the server may also be mobile, hence it may be difficult for a node to connect to the server. In addition, the server could be the operation bottleneck as it may be just a normal Ad-Hoc node with limited memory, CPU processing capacity and battery power. In order to address this problem, the position service system and the certificate authority should be distributed among a number of servers deployed in the network.

The need for scalable and energy efficient protocols, along with the recent availability of small, inexpensive and low power positioning instruments justify introducing position based routing algorithms in mobile Ad-Hoc networks. For the aforementioned reasons, it is a challenge to find a scalable, distributed and secure position-based routing protocol for Ad-Hoc networks. A new model of routing protocol, ARANz has been proposed in this work.

This paper (which is a continuation of our work in [1] and [2]) discusses the new protocol ARANz and compares it to *Ad-Hoc On-demand Distance Vector (AODV)* [3] and *Authenticated Routing for Ad-Hoc Networks (ARAN)* [4] protocols. The discussed protocols are compared with respect to their security, the used route discovery and path selection techniques, guaranteeing loop-freedom, the suitable network density to be implemented in, load distribution, the need of centralized trust and/or synchronization, robustness, implementation complexity, scalability, packet and processing overhead, route acquisition latency and data packets' end-to-end delay.

The rest of the paper is organized as follows. Section 2 looks at the existing and recent works on Ad-Hoc routing protocols. Section 3 presents our new routing protocol. Sections 4 and 5 contain a qualitative comparison as well as analysis and discussion of AODV, ARAN and ARANz protocols. We conclude our work in Section 6. Finally, we present our future direction in Section 7.

2.0 BACKGROUND

In this section we will discuss the existing and recent works on Ad-Hoc routing protocols. Subsections 2.2 and 2.3 give an overview about two particular protocol; AODV and ARAN protocols.

2.1 Existing Works

Several routing protocols have been proposed for mobile Ad-Hoc networks. In general, they can be divided into two main categories: ***topology-based*** and ***position-based***. ***Topology-based*** routing protocols use information about links that exist in the network to perform packet forwarding. They are, in turn, divided into three categories: ***proactive***, ***reactive***, and ***hybrid*** (hierarchical) protocols.

Proactive routing protocols periodically broadcast control messages in an attempt to have each node always know a current route to all destinations, and remove local routing entries if they time out. We observed that proactive routing protocols are less suitable for Ad-Hoc wireless networks because they constantly consume power throughout the network, regardless of the

presence of network activity. Also they are not designed to track topology changes occurring at a high rate [5][6].

On the other hand, ***reactive*** routing protocols are deemed more appropriate for wireless environments because they initiate a route discovery process only when data packets need to be routed. Many Ad-Hoc routing protocols that use reactive route determination have been developed such as AODV protocol. One advantage of reactive routing protocols is that no periodic routing packets are required. However, they may have poor performance in terms of control overhead in networks with high mobility and heavy traffic loads. Scalability is said to be another disadvantage because they rely on blind broadcasts to discover routes [6].

As seen, proactive routing uses excess bandwidth to maintain routing information, while reactive routing involves long route request delays. Reactive routing also inefficiently floods the entire network for route determination. ***Hybrid*** routing protocols, such as *Zone Routing Protocol (ZRP)* [5], aim to address these problems by combining the best properties of both approaches. The disadvantage of ZRP is that for large values of routing zone the protocol can behave like a pure proactive protocol, while for small values it behaves like a reactive protocol [7].

In general, topology-based are considered not to scale in networks with more than several hundred nodes [8]. We note that none of the Ad-Hoc routing protocols mentioned above defined their security requirements and that they inherently trust all participants. Obviously, this could result in security vulnerabilities and exposures that could easily allow routing attacks [4][9][10].

Since then, many works were done on secure routing protocols such as ARAN, *Secure Ad-Hoc On-demand Distance Vector (SAODV)* [11] and *ARIADNE* [12]. Of particular is the ARAN protocol. Effectively, ARAN is similar to AODV, but provide authentication of route discovery, setup, and maintenance. The main objectives of ARAN are to detect and protect against attacks from malicious nodes in a managed-open environment where no network infrastructure is pre-deployed, however it expects a small amount of prior security coordination. It also requires the use of a trusted certificate Authority server. In comparison to basic AODV, ARAN prevents a number of attacks such as modification, impersonation and fabrication exploits. We observed that although ARAN has good and equivalent performance to AODV, its route discovery process results in more packet overhead and higher latency since each packet must be signed. ARAN is also based on a centralized trust, hence, suffers from the compromised server problem and the single point of failure. ARAN does not scale well in large networks since any request packet is broadcasted to all nodes in the network.

In recent developments, ***position-based*** routing protocols exhibit better scalability, performance, and robustness against frequent topological changes [8][13]. Position-based routing protocols use the geographical position of nodes to make routing decisions, which results in improving efficiency and performance. These protocols require that a node be able to obtain its own geographical position and the geographical position of the destination. Generally, this information is obtained via Global Positioning System (GPS) and location services. There are different kinds of position-based protocols that are categorized into three main groups: ***Restricted directional flooding***, ***Greedy*** and ***hierarchical*** routing protocols.

Most position-based protocols, such as *Greedy Perimeter Stateless Routing (GPSR)* [14], use ***greedy*** forwarding to route packets from a source to the destination. In greedy forwarding, a source node selects a neighboring node that is closest to the destination as the next hop.

Similarly, each intermediate node selects a next hop node until the packet reaches the destination. In order to enable the nodes to do this, nodes periodically broadcast small packets (called beacons) to announce their position and enable other nodes to maintain a one-hop neighbor table. Such an approach is scalable since it does not need routing discovery and maintenance [15]. However, periodic beaconing creates a lot of congestion in the network and consume the nodes' energy [8][13]. In addition, Greedy forwarding in general may not always find the optimum route [15]. GPSR for example works well in dense networks, but in sparse networks greedy forwarding fails due to voids [14].

Location-Aided Routing (LAR) [16] is an example of ***restricted directional flooding*** routing protocols in which , the sender will broadcast the packet to all single hop neighbors towards the destination. In the LAR approach, the node which received the route request message, compares its distance to the destination, with the distance of the previous hop to the destination. If the receiver node is closer to the destination, it retransmits the route request message; otherwise, it will drop the message. In order to find the shortest path in the network level, instead of selecting a single node as the next hop, several nodes will be selected for managing the route request message and each of them will put its IP address in the header of the request packet (this will increase the size of the message). Therefore, the route through which the route request message is passed will be saved in the header of the message.

TERMINODES [17] is an example of ***hierarchical*** routing protocols. TERMINODES presents a two level hierarchy within which, if the destination is close to the sender (in number of hops), packets will be routed based on a proactive distance vector. Greedy routing is used in long distance routing.

All the aforementioned position-based routing protocols are vulnerable to various security attacks since they were not designed with security in mind [10]. With the exceptions of LAR, they have low probability to find the shortest path.

Few secure position-based routing protocols have been proposed such as *Secure Position Aided Ad-Hoc Routing (SPAAR)* [18], *Anonymous On-Demand Position-based Routing in Mobile Ad-Hoc Networks (AODPR)* [10], and *Secure Geographic Forwarding (SGF)* [19]. However they suffer from some problems; such as, the single point of failure and attack, increased packet and processing overhead, and/or scalability problems.

From observations, we note that despite its popularities, many topology-based routing protocols still possess security vulnerabilities and are not scalable. Although some improvements on security aspects were proposed such as in ARAN, the implicit trust on centralized node has introduced other security problems. Like the others, ARAN does not scale well. Finally, restricted directional flooding has better performance than topology-based and other position-based routing protocols.

2.2 Overview of AODV Protocol

Ad-Hoc On-demand Distance Vector (AODV) [3] is classified as a pure on-demand route acquisition protocol, since nodes that are not on a selected path do not maintain routing information or participate in routing table exchanges. AODV offers a quick adaptation to dynamic link conditions, low processing, memory overhead, and low network utilization. It uses destination sequence numbers to ensure loop freedom at all times.

When a node requires a route to a destination, if the route is not available, the node initiates a route discovery process within the network. It broadcasts a Route Request Packet (RREQ) to its neighbors. Upon receipt of RREQ, the node creates a reverse routing entry towards the originator of RREQ, which is used to forward replies later.

Once the request reaches the destination or an intermediate node with a fresh enough route, the destination or intermediate node responds by unicasting a Route REPly packet (RREP) back to the neighbor from which it first received the request. Upon receipt of RREP, the reverse routing entry towards the originator of RREP is also created, similar to the processing of RREQ. A precursor list is associated with each routing entry, which is created at the same time. The precursor list contains the upstream nodes towards the same destinations.

For route maintenance; every node along an active route periodically broadcasts HELLO messages to its neighbors. If the node does not receive a HELLO message or a data packet from a neighbor for a while, the link between itself and the neighbor is considered to be broken. If the destination is not far away (from the invalid routing entry), local repair mechanism may be launched to rebuild the route towards the destination; otherwise, a REER (Route Error) packet is sent to the neighbors in the precursor list associated with the routing entry to inform them of the link failure.

2.3 Overview of ARAN Protocol

Effectively, *Authenticated Routing for Ad-Hoc Networks (ARAN)* [4] protocol is similar to AODV. However, the former provides authentication of route discovery, setup, and maintenance. The main objectives of ARAN are to detect and protect against attacks from malicious nodes in a managed-open environment where no network infrastructure is pre-deployed, however it expects a small amount of prior security coordination. It requires the use of a trusted Certificate Authority (CA) server whose public key is known by all valid nodes. Before entering the Ad-Hoc network each node requests a certificate from this CA. ARAN uses cryptographic certificates to prevent and detect most of the security attacks that most of the ad hoc routing protocols face. This protocol introduces authentication, message integrity and non-repudiation as part of a minimal security policy for the Ad-Hoc environment.

ARAN consists of a preliminary certification process followed by a route instantiation process that guarantees end-to-end authentication. Thus, the routing messages are authenticated end-to-end and only authorized nodes participate at each hop between source and destination.

Route discovery in ARAN is accomplished by broadcasting a Route Discovery Packet (RDP) from a source node which is replied to by a unicast REPly (REP) packet that is launched from the destination and sent back along the reverse path to the source. The routing messages are authenticated at each hop from source to destination, as well as on the reverse path from the destination to the source. Hence, every node that forwards a request or a reply must also sign it so that the following node can check the validity of the previous node. Because only the destination can send REPs, loop freedom is guaranteed easily.

ARAN requires that nodes keep one routing table entry per source-destination pair that is currently active. This is certainly more costly than per-destination entries in non-secure Ad-Hoc routing protocols.

Although there is a greater performance cost to ARAN as compared to AODV, the increase in cost is minimal and outweighed by the increased security. Compared to basic AODV, ARAN prevents a number of attacks, including spoofing of route signaling messages, alteration of routing messages and replay attacks. Moreover, simulation results in [4] show that ARAN has a good performance, equivalent to AODV, in discovering and maintaining routes. On the other hand, besides its problems in handling scalability with the number of nodes (that are inherited by AODV) it causes more packet overhead and higher latency in route discovery since each packet must be signed. Finally, ARAN uses one certificate server and this leads to an extreme need to keep this server uncompromised.

3.0 PROPOSED PROTOCOL

In this section, we propose a new routing model called ARANz. The proposed protocol was called ARANz since it adopts the authentication steps used with the ARAN protocol and deals with the network as zones.

3.1 Introduction

ARANz, just like ARAN, uses cryptographic certificates to prevent most of the attacks against Ad-Hoc routing protocols and detect erratic behavior. However, ARANz introduces a hierarchal distributed routing algorithm, which aims to improve performance of the routing protocol and distribute load by dividing the area into zones. Moreover it tries to achieve robustness and high level of security, solve the single point of failure problem and avoid single point of attack problem by distributing trust among multiple Local Certificate Authority (LCA) servers. Each zone has multiple LCAs that should collaborate with each other to issue certificates for the nodes inside that zone and work as backups of each other. If a misbehavior detection scheme is present on the network, then the security of the protocol can be improved through collaboration with this scheme.

Moreover, ARANz tries to exhibit better scalability, performance, and robustness against frequent topological changes by taking advantage of the idea of restricted directional flooding position-based routing protocols. Whenever a node needs to communicate with another , the former will get the latter's position through the LCAs of its zone, then the route request packet is sent using restricted directional flooding. This helps in reducing overall overhead and saving network bandwidth. Hence, the LCAs work also as Position Servers; and each node should tell the LCAs of its zone about its new position if it has moved at a rate proportional to its speed.

ARANz consists mainly of five phases which are network setup, network maintenance, location service, route instantiation and maintenance and finally data transmission. Network setup includes certifying trusted nodes, dividing area into zones and electing initial certificate authority servers. Network maintenance phase copes with ensuring maintenance of the network structure taking into consideration some issues like updating nodes' certificates, LCAs synchronization, movements of nodes in and out the network as well as corrupted and destroyed nodes.

Whenever a node has data to be sent to a particular destination; it is supposed to obtain the destination's position before beginning the route discovery process. Location service phase enables the source to obtain the destination's position via communicating LCAs in its zone. After getting the destination's position route, the instantiation and maintenance phase is initiated. The source begins route discovery to destination by sending a Route Discovery Packet (RDP). This is

done using restricted directional flooding towards the destination node. Upon receiving the first RDP, destination unicasts a Route REPly (RREP) packet back along the reverse path to the source to setup the route. After finishing route discovery and setup the source begins sending the data to the destination. In order to maintain the selected route, nodes in ARANz keep track of whether routes are active or not and use ERRor (ERR) packets to report links in active routes that are broken due to node movement.

Since each node by the end of the network setup phase has its own certificate, these certificates can be used to apply the authentication steps used with ARAN protocol. Hence the source of any packet and all intermediate nodes sign the packet using their private keys and append their certificates to the packets. Also, each intermediate node, as well as the destination, validates the previous node's signature using the previous node's public key which is extracted from its certificate. Thus, it is assured that packets sent during the route discovery are authenticated end-to-end and only authorized nodes participate at each hop between source and destination. Consequently, as in ARAN, data packets exchanged between nodes are not signed and do not have attached certificates. Hence, each node simply relays data packets to its successor in the route obtained during the route initiation process. Fig. 1 shows the general flowchart of our proposed protocol.

3.2 Important Assumptions

We assume (Nn) cooperative nodes in a managed-open environment. These nodes are distributed randomly in (Ar×Ar) Km^2 area and are aware of their positions (equipped with GPS receivers). This area will be divided into (Nz) zones; the area of each zone is (Ar×Ar)/Nz km^2. Communication among nodes is done mainly using restricted directional flooding adopting the authentication steps used in the ARAN protocol. A particular node in the network is chosen to have the software needed to begin the network setup, divide the area into zones and elect the initial LCAs. This node is called the Primary Certificate Authority (PCA) server and has the private part of the network key (K_{NET-}). All the trusted nodes that will participate in the network have a private/public key pair, the public part of the network key (K_{NET+}) and a Common Key (CK) which is used for encryption and decryption of the packets sent by non-PCA nodes in the network setup phase. In managed-open environments, keys are a priori generated and exchanged through an existing relationship between PCA and each trusted node.

3.3 Network Setup

The PCA starts the network setup by broadcasting a packet notifying the nodes of the beginning of the NETwork SETup (NETSET). This packet is signed by K_{NET-} to enable nodes to make sure that the PCA is actually the node that has sent the packet. Each node found in the network, upon receiving the first NETSET packet will record the IP address of the previous node, continue broadcasting the packet and reply with a Node INformation (NIN) packet to the PCA containing the node's IP address (IP_A), along with the needed information to elect the LCAs. The NIN packets are encrypted using the CK. Each node upon the receipt of a NIN packet will try to decrypt it using CK to ensure that its previous node is trusted and to proceed in processing the packet; otherwise the packet is dropped. After encrypting the NIN packet, it is sent through the reverse path until it reaches the PCA.

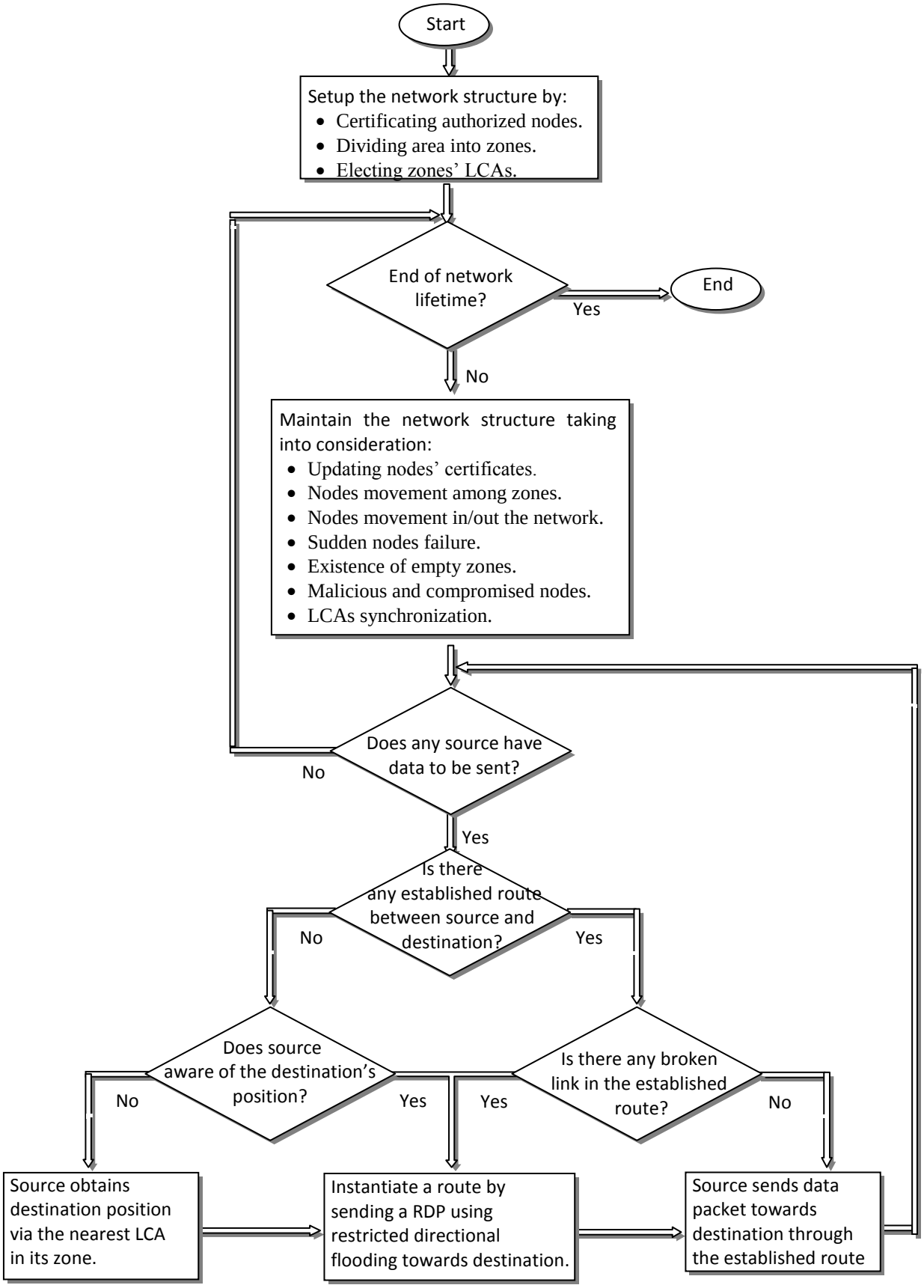

Fig. 1: System flowchart

After receiving the NIN packets from all authorized nodes existing currently in the network, PCA will divide the network into multiple equal-sized virtual zones and assign four LCAs for each zone. These LCAs are chosen to be on the zones' boundaries to make communication between LCAs of different zones easier and faster. The network structure is shown in Fig. 2, if we suppose that the whole area is divided, for example, into nine zones.

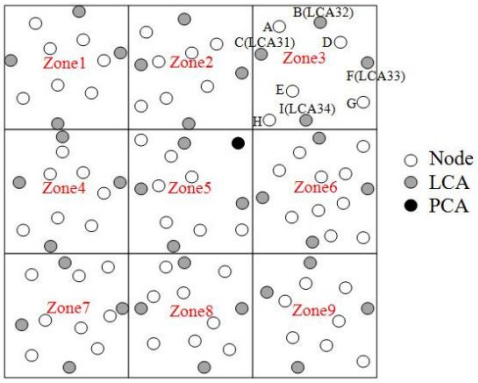

Fig. 2: Network structure

After that, the PCA will unicast a Node ROLE message (NROLE) to each participant node. Source routing will be used to send these messages since the PCA knows the position of all the nodes in the network. These messages will enable each node to know its role in the network (LCA or regular node).

Hence, the PCA will unicast a NROLE message for each participating regular node *n* containing node's certificate ($Cert_n$), number of the zone where it resides (z), identities and positions of LCAs in its zone ($LCAsZ_z$), and the public key that will be used in this zone (K_{Zz+}). The node certificate ($Cert_n$) contains the IP address of *n* (IP_n), the public key of *n* (K_{n+}), a timestamp (t) of when the certificate was created, and a time (e) at which the certificate expires. These variables are concatenated and signed with the K_{NET-}. Nodes use these certificates to authenticate themselves to other nodes during the exchange of network maintenance, position and routing packets.

The PCA also will unicast a NROLE message for each LCA containing the node's certificate, zone LCAs certificate ($CertLZ_z$), the number of that LCA in its zone, the number and coordinates of the zone it is responsible for, numbers and coordinates of this zone's 8-neighboring zones ($8NbrZ_Z$), private/public key pair that will be used in this zone, identity and position of other LCAs in this zone ($LCAsZ_z$), identity and position of its adjacent LCA in the neighboring zone, public key and part of the private key of the immediate neighboring zone (will be used in the case that neighboring zone became empty), and the authentication table. Moreover it will contain a list of IP addresses and public keys of authorized nodes that were not in the network during network setup (Absent_Nodes); this will enable these nodes to join the network from any zone at any time.

The authentication table contains a tuple (IP address, public key, time stamp (t), certificate expiration date (e), and position) for each node that is in this zone. It is used to update the nodes' certificates. Also it is used upon receiving a position request packet; LCA checks whether the

destination of the route is local or external one; in order to send a position reply packet to the source or send position request packet to adjacent zone respectively.

The zone LCA certificate ($CertLZ_z$) binds the zone's number to its public key and contains the zone number, zone public key, time stamp and Certificate expiration date. These certificates are signed by the zone private key and used by LCAs as a proof that they are LCAs of the specified zone. These Certificates are used between LCAs of different zones and between LCAs and nodes in their zones during the exchange of network maintenance and position packets.

3.4 Network Maintenance

After the network setup phase, the node can update its certificate, move freely in the network, move in and out the network, become corrupted or even destroyed, etc. our protocol should be able to cope with these issues.

Since each node by the end of the network setup phase will have its node certificate, these certificates can be used to apply the authentication steps used with ARAN protocol. Hence the source of any packet will sign the packet using its private key and appends its node certificate to the packet. If the source of a packet is an LCA it will also include its zone LCA certificate within the packet to enable the destination to make sure that the LCA has a valid certificate for a particular zone. Each node along the path validates the previous node's signature (using the previous node's public key, which is extracted from its certificate), removes the previous node's certificate and signature, signs the original contents of the packet, and appends its own certificate.

Another important thing to be mentioned is that the packets sent from the nodes to LCAs of their zones is done using restricted directional flooding, since each node within that zone knows the position of these LCAs. Also communication between nodes (in the same zone or different zones) is done using restricted directional flooding (after obtaining the destination position by the source). Restricted directional flooding is also used for communications among adjacent LCAs in neighboring zones (if they are not reachable within one hop). However source routing is used to send packets among LCAs of the same zone and from the LCAs to nodes in their zones; since these LCAs know the position of all the nodes in their zone.

By default, reply packets are sent through reverse paths of their corresponding request packets. Finally, to circumvent voids (regions without nodes) in sparse networks; if the restricted directional flooding of a request fails after three attempts, the packet is broadcasted to the whole network.

3.4.1 Certifications Update

All nodes in a specific zone must maintain valid certificates with the LCAs in their zone. This is done by periodically sending a Certificate REQuest (CREQ) packet to any one of these LCAs, however; each node may update its certificate from the nearest LCA to itself to reduce overhead. This CREQ packet is signed by the node's private key and sent using restricted directional flooding. Fig. 3, shows the certificate request packets sent for updating Node K's certificate.

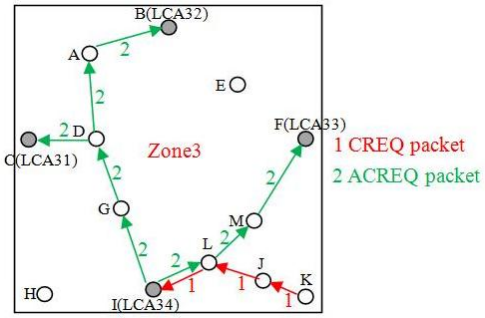

Fig. 3: Node K's certification update

Each intermediate node that receives this CREQ packet sets up a reverse path back to the source by recording the neighbor from which it received the packet. This is in anticipation of eventually receiving a certificate reply packet that it will need to feedback to the source. The receiving node uses public key of the previous hop, which it extracts from its certificate, to validate the signature and verify that its certificate has not expired. The receiving node also verifies that it has not already processed this CREQ. The receiving node signs the contents of the packet, appends its own certificate, and forward broadcasts the packet to its neighbors.

The corresponding LCA upon receiving the first CREQ packet will communicate other LCAs in its zone to ask them whether to update this certificate or not. This is done by sending a packet to each LCA asking for Acceptance of the Certificate REQuest (ACREQ). So a specific LCA will be allowed to issue a certificate only if it receives Acceptance for Certificate REPly (ACREP) packet from the majority of the LCAs of that zone (signed by their private keys). This will help in increasing the robustness and security of the protocol; if one server fails or is compromised the other three servers will still be able to issue valid certificates.

In the case that there are no failed or compromised servers, the LCA will be allowed to issue a fresh certificate for the node after receiving at least three ACREP packets (one of them may be from itself). Then the LCA will unicast a Certificate REPly (CREP) packet back along the reverse path to the source. It also unicasts a Node CERTificate (NCERT) packet to other LCAs in its zone containing the new issued certificate (since LCAs inside a specific zone carry identical information).

The LCAs also must maintain fresh node and zone LCA certificates. Hence periodically each LCA should unicast ACREQ to other LCAs in its zone. Upon receiving the ACREPs it will be issued both node and zone LCA certificates.

3.4.2 Nodes Mobility

If a regular node has moved a pre defined distance (d_{mov}) from its last known position it should include its new position in the CREQ packet sent to the nearest LCA in its zone. This LCA will in turn send the node's position to other LCAs in its zone within the ACREQ packet. This will help the LCAs keep track of up-to-date positions of the nodes inside the zone and enable them to discover that a specific node has departed this zone to the neighboring one.

If the node leaves to one of the immediate 4-neighboring zones, the LCAs of the departed zone will remove the node's information from their tables and the nearest LCA to the new zone will send a Departed NODE (DNODE) packet to its adjacent LCA. This packet indicates that this

node is trusted and contains the node's position. Fig. 4 shows the communication done when R leaves zone number 5 to zone number 6 (moves from position P_R to P'_R).

The LCA in the new zone will send a New ZONE (NZONE) packet to the departing node; containing the number and public key of the new zone, in addition to IP addresses and positions of LCAs of that zone. This LCA also will send multiple New NODE (NNODE) packets to other LCAs in its zone informing them about the new node.

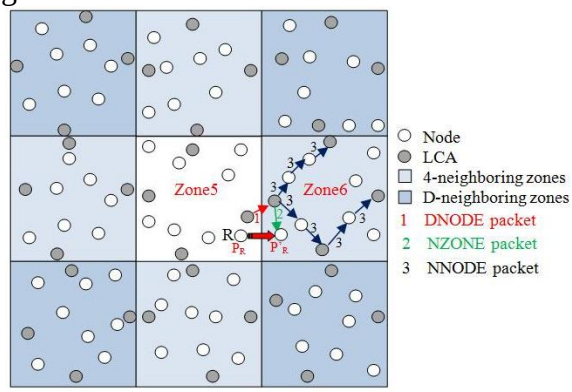

Fig. 4: Movement of node R from zone 5 to a 4-neighboring zone

However if the node leaves to one of the diagonal D-neighboring zones, the LCA of the original zone will send a DNODE packet to the adjacent LCA in the immediate neighboring zone to indicate that this node is trusted. This LCA in turn resends the packet to the LCA adjacent to the new D-neighboring zone. The latest will resend this packet to the adjacent LCA in its immediate neighboring zone. Now the LCA in the neighboring zone that receives the packet will send an NZONE packet to the departing node. This LCA also will send multiple unicasts to other LCAs in its zone telling them about the new node. Fig. 5 shows the DNODE and NZONE packets sent when node R leaves zone number 5 to zone number 9 (moves from position PR to P'R).

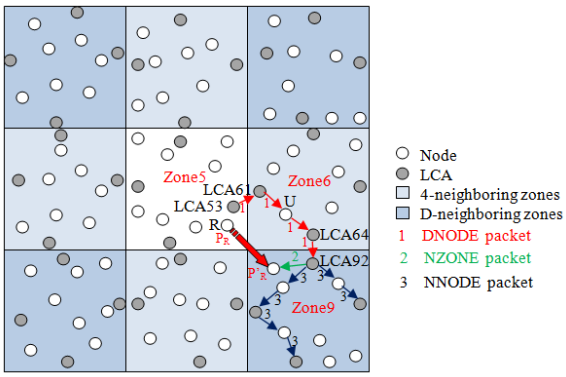

Fig. 5: Movement of node R from zone 5 to a D-neighboring zone

If any LCA has been moved the pre defined distance (d_{mov}) from its last known position, it must broadcast its position to the nodes inside its zone (including other LCAs). It also should send its position to its adjacent LCA in the neighboring zone. However, an LCA may decide to leave its zone, or its distance from the middle point of the zone side may become higher than a pre defined distance (d_{sid}). In these two cases a new LCA election is required. Upon deciding to leave its zone, the LCA will send a New LCA Election (NLCAE) packet to nodes in its zone. Each node in the corresponding zone will calculate its probability by itself to reduce the load on the leaving CA. Then each node will send its calculated probability, through reverse path, to the leaving LCA. Now the leaving LCA selects the node with the highest probability to become the

new LCA. Then it broadcasts a New LCA (NLCA) packet so that all nodes inside that zone know the address and position of the new LCA. This information is also sent to the adjacent LCA in the neighboring zone through a New Adjacent LCA (NALCA) packet. Now leaving LCA transfers to the new LCA the needed information (similar to that included in the NROLE message sent from PCA to LCA nodes during network setup phase).

3.4.3 Nodes Failure

The sudden failure of an LCA (or nodes movement outside the network boundaries) can be discovered from the periodic LCA zone and node certificates update of LCAs. Hence, if the LCAs in a particular zone did not receive the ACREQ packet from a specific LCA in a pre determined time they will discover that this LCA has a problem. So, one of these LCAs should take the responsibility of electing a new LCA and broadcasting NLCA and NALCA packets similar to those sent in the case of LCA nodes mobility.

Subsequently, if the failed LCA has been repaired, it will come back to the network as a regular node. To enable this node to join the network from any zone, node's IP address and public key will be sent to all LCAs in the network. Hence each LCA in zone number 5 will send a Failed NODE (FNODE) packet to its adjacent LCA in the neighboring zone. The later in turn will send it to LCAs in its zone, and so on.

Regular nodes failure also can be discovered from the periodic node certificate update. If an LCA had in its Authentication table an expired node certificate, and did not receive a CREQ packet within a predefined period of time it will discover that this node has a problem. Then the LCA that had issued the last certificate for that node will send a FNODE packet.

3.4.4 Empty Zones

Due to nodes movement, some zones may become empty. When many nodes leave a specific zone, the last four nodes stay in that zone will be its four LCAs. If any one of these LCAs wants to leave the zone, it should transfer its responsibilities to one of the other LCAs. This will continue until the last node in the zone (that plays the role of the four LCAs) decides to leave the zone. Upon departing its zone it will send a packet to its adjacent LCA in the zone it is leaving for. This packet informs the LCA of the new zone that this node is the last node leaving the zone. This Empty ZONE (EZONE) packet will be sent to the 8-neighboring zones (4-neighboring zones and D-neighboring zones) of the empty zone informing them that this zone is empty.

Now let us assume that a node leaves a specific zone and enters the empty zone. The LCA of the departed zone knows that this zone is empty, so it will send a packet to the other immediate neighboring zones of the empty zone asking them to send the part of the empty zone private key that they have. The LCA of the departed zone, upon receiving these parts will combine them and send a packet to the new node informing it that it is the only node in the zone and giving it the needed information. The new node will issue to itself the needed certificates and play the role of the four LCAs of the zone until other nodes enter. For example, if another node enters this zone each one of them will play the role of the two LCAs according to their positions, and so on.

3.4.5 Compromised Nodes

Our protocol can collaborate with a misbehavior detection system. If regular nodes detect misbehavior of other nodes, they will send Misbehavior NODE (MNODE) packets to report this to the LCAs of their zones. If the majority of the LCAs in a particular zone have received a predefined number of MNODE packets for the same node then they can collaborate and broadcast a Compromised NODE (CNODE) packet. So other nodes will exclude this node from the routes until its certificate expires normally.

The same technique can be used if the LCAs of a zone detect any misbehavior of a specific LCA in that zone; i.e., if three LCAs in a specific zone have detected that the fourth LCA has misbehavior actions, they will remove this LCA from the $LCAsZ_z$, broadcast a CNODE packet and initiate a new LCA election process.

3.4.6 Malicious Nodes

Malicious nodes may cause some erratic behaviors such as the use of invalid certificates, improperly signed packets, and misuse of some packets. ARANz's response with regards to all erratic behaviors in the same way; dropping all packets that has any erratic behavior.

3.4.7 LCAs Synchronization

All the LCAs in the network should have synchronized clocks to ensure the correctness of the protocol; to avoid a situation such that two nodes in different zones (or even in the same zone) are issued certificates at the same moment with two different time stamps. Hence, the type of synchronization needed for our protocol is maintaining relative clocks rather than having the clocks synchronized (adjusted) to a reference clock in the network; i.e., nodes run their local clocks independently, but keep information about the difference between their clocks and the system's clock so that at any instant the local time of the node can be converted to the system's time.

As a starting point, the PCA may include a time stamp within the NROLE message sent to the LCAs during the network setup phase. So each LCA will be able to know the difference between its local clock and the LCA's clock. Also, a time stamp may be included in the information sent to a new elected LCA.

Moreover, all clocks are subject to clock drift; as oscillators' frequency will vary unpredictably due to various physical effects [20]. Hence, periodically one of the LCAs may send a message containing a time stamp to other LCAs in the network to eliminate the effect of LCAs' clocks drifts. In order to increase the robustness of the system, the LCAs will alternate this job. Also a nonce is used to avoid replay attack. Certainly, the LCA includes its zone LCAs certificate within the message, signs the contents of the message, and appends its own certificate. These packets are sent among the LCAs in the same way as the Position REQuest packets (section 3.5).

Regular nodes can use the timestamp included in its certificate to know the system's time and check the validity of the certificates of other nodes; so there is no need for extra communications between the LCAs and the regular nodes in a specific zone.

3.5 Location Service

Before beginning the route discovery the source should know the destination's position. The source (S) sends a Position REQuest (PREQ) Packet to the nearest LCA in its zone using restricted directional flooding to ask the LCA about the position of the destination (D) (refer to Fig. 6).

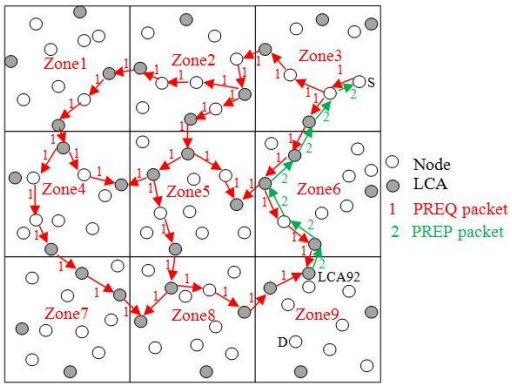

Fig. 6: Authenticated location service

Upon receiving the first PREQ the LCA will check whether the destination is in its zone or not. If the destination is in the same zone of the source, the destination will be found in the authentication table of the LCA. Hence the LCA will unicast a Position REPly (PREP) Packet to the source. This PREP contains the destination's position and goes back along the reverse path to source.

If the destination is in a different zone, the destination will not be found in the authentication table of the LCA. So the LCA will send multiple unicast PREQ (using source routing) to the other LCAs in its zone that have adjacent LCAs in neighboring zones. Each LCA in that zone will send this PREQ to its adjacent LCA in the neighboring zone. Now each LCA in the neighboring zones will check if it has received the packet from other LCAs in its zone, and it will drop it. Else, it unicasts PREQ to the other LCAs in its zone that have adjacent LCAs in the neighboring zones. These steps will be repeated until one of the LCAs (LCA92 in Fig. 6) finds the destination in its authentication table. This LCA, in turn, will unicast a PREP back along the reverse path to source.

All position discovery steps are done using the authentication steps used with ARAN protocol.

3.6 Route Discovery, Setup and Maintenance

After getting the destination's position (whether local or external one) the source begins route instantiation to destination by sending a Route Discovery Packet (RDP). This is done using restricted directional flooding to the source's neighbors. When the destination receives the first RDP it unicasts a Route REPly (RREP) Packet back along the reverse path to the source. All the route discovery steps are done using the authentication steps used with ARAN protocol.

ARANz is an on-demand routing protocol; nodes keep track of whether routes are active or not. When no data is received on an existing route for that route's lifetime, the route is simply deactivated. Data received on an inactive route causes nodes to generate an ERRor (ERR) packet. Nodes also use ERR packets to report links in active routes that are broken due to node

movement (as AODV and ARAN, ARANz uses hello messages to the neighbors in order to detect possible link failure). All ERR packets must be signed.

3.7 Data Transmission

After finishing the route discovery and setup the source will begin sending the data to the destination. As in ARAN, only the control messages between nodes are subject to signing and verifying; once the route reply reaches the originator, it is guaranteed that the route found is authentic. Thus, data packets exchanged between nodes after a route has been set up are not processed by ARANz in any way; they do not have attached certificates and are not signed. Once a route is set up, the routing daemon is out of the picture until that route becomes invalid and is needed again. However, to ensure data privacy and prevent other trusted nodes from reading the data itself, the data may be encrypted using the public key of the destination which the source may obtain through the position discovery phase.

4.0 COMPARISON OF PRESENTED PROTOCOLS

Table.1 summarizes the discussed protocols together with the evaluation criteria used. This summary is a high level qualitative comparison of the protocols rather than a precise quantitative performance evaluation.

As discussed earlier both AODV and ARAN are reactive topology-based routing protocols that use broadcasting in the route discovery process; while ARANz is a restricted directional flooding position-based routing protocol. AODV does not define any security requirements and inherently trusts all participants. On the other hand, ARAN and ARANz use cryptographic certificates to prevent most of the attacks against Ad-Hoc routing protocols and detect erratic behavior. Both ARAN and ARANz achieve authentication, confidentiality, integrity, non-repudiation, anti-spoofing as part of a minimal security policy for the ad hoc environment. ARANz, moreover, tries to achieve a higher level of security and avoid the single point of attack problem by distributing trusts among multiple LCAs. All of the three protocols are loop-free and hence preserve the network resources and guarantee the correct operation of the protocol. All of them also may be implemented on any network density.

Table.1 Characteristics of the presented protocols.

Protocols / Performance Parameters	AODV	ARAN	ARANz
Type	Topology-based (Reactive)	Topology-based (Reactive)	Position-based (Restricted Directional Flooding)
Secure	No	Yes	Yes
Route Discovery	Broadcasting	Broadcasting	Restricted Directional Flooding
Path Selection	Least number of hops	Quickest	Quickest
Loop Freedom	Yes	Yes	Yes
Density	All	All	All
Load Distribution	Yes	No	Yes
Centralized Trust	No	Yes (Certificate Authority)	No
Synchronization	No	No	Yes
Robustness	Medium	Medium	Medium
Implementation Complexity	Low	Medium	High
Scalability	Medium	Low	High
Packet Overhead	Medium	High	Medium
Processing Overhead	Low	Medium	Medium
Route Acquisition Latency	Low	High	Medium
Data Packets' End-to-End delay	Medium	Medium	Medium

AODV selects the path with the minimum number of hops. ARAN and ARANz do not guarantee the shortest path, but they offer the quickest path which is chosen by the RDP that reaches the destination first. Simulations in [4] showed that the average path length for the AODV and ARAN are almost identical. This indicates that even though ARAN does not explicitly seek shortest paths, the first route discovery packet to reach the destination usually travels along the shortest path. Hence ARAN is as effective as AODV in finding the shortest path. It is expected for ARANz to be the same.

In ARAN each node should update its certificate from the trusted CA server; hence the load is centralized on that CA. This CA also presents a centralized trust and thus may be the system's single point of attack. ARANz, however, tries to distribute load and trust by dividing the area into zones and introducing multiple LCAs in each zone. Thus, compromising one LCA will not prevent other LCAs from updating the certificates and electing a new LCA to replace the compromised one. Using multiple LCAs in ARANz, on the other hand, results in the need to keep them synchronized.

AODV and ARAN are more robust in the route discovery phase than ARANz since they broadcast the route request to the whole network. ARANz however uses restricted directional flooding to discover routes and this may increase the effect of a failure or movement of a single node. After setting up the route, the three protocols have the same robustness since the failure of an individual node might result in packet loss and the setting up of a new route. ARANz tries to achieve higher robustness compared to ARAN by distributing trust among different LCAs; multiple LCAs collaborates to issue certificates for the nodes inside a particular zone and work as backups of each others.

Hence the failure of a single LCA (or even multiple LCAs) will not affect the update of the certificates. However in ARAN the failure of a single node (CA) will prevent all the nodes from updating their certificates. After taking these points into consideration the robustness of AODV is considered high and those of ARAN and ARANz are considered as medium.

Implementation complexity describes how complex it is to implement and test a particular protocol. This measure is highly subjective and we will explain our opinion while discussing each protocol. It is very easy to implement the AODV protocol since it is not secure and simply broadcasts the RREQ packet to all nodes in the network. ARAN is considered to have medium implementation complexity due to certification update and encryption/decryption of the messages. Lastly, ARANz has the highest implementation complexity due to its security, dealing with the network as zones and introducing multiple LCAs in each zone.

Scalability describes the performance of the protocol with increasing number of nodes in the network. The scalability of AODV is considered as medium since its approach can handle networks with a reasonable size, but may have problems if it grows due to broadcasting RREQ packets. ARAN may have worse performance than AODV in large networks. ARAN assumes the existence of one certificate server, which may be the operation bottleneck especially in large area networks. Moreover, increasing the number of nodes in the network by using broadcasting will increase the packet overhead. Finally, in large area networks the probability of having long routes will increase, and since each node spends time in the encryption/decryption of the messages, the probability of node movements and route breakage will increase. For these three reasons ARAN is considered to have a low scalability.

ARANz has high scalability since it will be able to maintain good performance even with large networks. This is due to using restricted directional flooding instead of broadcasting, dividing the area into zones and distributing load among multiple LCAs. Messages related to location service should not highly affect scalability since all of these messages are sent using source routing or restricted directional flooding. Even LCA election process is done by broadcasting the NLCAE packet to nodes in the intended zone only.

Packet overhead refers to bandwidth consumption due to larger packets and/or higher number of signaling packets. AODV has a medium packet overhead because of its small-size packets compared to ARAN and ARANz, and increased number of packets due to broadcasting. ARAN has a high packet overhead because of the large-size packets due to certificates and signatures stored in packets and increased number of packets due to broadcasting. ARANz has a medium packet overhead because of the large-size packets due to the security techniques used and decreased number of packets compared to AODV and ARAN due to using restricted directional flooding. Location service messages should not significantly affect packet overhead (especially if the source and destination are in the same zone) since all of these messages are sent using source routing or restricted directional flooding. Even LCA election process and certificate updates are done locally.

Processing overhead is used to associate each protocol with processing requirements. AODV requires a low CPU processing since it is an unsecure protocol. ARAN and ARANz, on the other hand, have medium processing overhead due to validating the previous node's signature, removing the previous node's certificate and signature, signing the original contents of the packet, and appending the nodes' certificate.

Route acquisition latency is an indication of the delay between the sending of a route request/discovery packet by a source for discovering a route to a destination and the receipt of the first corresponding route reply. Simulations in [4] show that the average route acquisition latency for ARAN is approximately double that for AODV due to ARAN′s cryptographic operations. It is expected that ARANz will have lower route acquisition latency than ARAN. Since ARAN broadcasts the RDP packet, processing RDP packet of other route discovery processes by a specific node is delayed until this RDP packet is processed; i.e., increasing other routes' acquisition latencies. ARANz however may limit this problem due to using restricted directional flooding. Thus the route acquisition latency is considered as low for AODV, high for ARAN, and medium for ARANz.

End-to-end delay of data packets is the delay between the sending of the data packet by the source and its receipt at the corresponding receiver. This includes all the delays caused during position discovery, route acquisition, buffering and processing at intermediate nodes, and retransmission delays at the MAC layer. One may expect that AODV has the lowest end-to-end delay since it is not secure, and that ARAN has medium end-to-end delay due to its cryptographic operations. Also, it is expected that the highest end-to-end delay is for ARANz due to the position discovery process done before performing the authenticated route discovery. However, simulations in [4] shows that the end-to-end delay of data packets for the AODV and ARAN protocols are almost identical. Although ARAN has higher route acquisition latency, the number of route discoveries performed is a small fraction of the number of data packets delivered. Hence, the effect of the route acquisition latency on average end-to-end delay of data packets is not significant. The processing of data packets is identical when using either protocol,

and so the average latency is nearly the same. So the end-to-end delay of data packets is considered medium for the three protocols.

5.0 ANALYSIS AND DISCUSSION

The three discussed protocols are loop-free, may be implemented at any network density, effective in finding the shortest path, and have almost identical end-to-end delay of data packets.

AODV is a non-secure reactive routing protocol; which reduces its processing overhead. It uses broadcasting in the route discovery phase which increases its robustness against nodes' failure during this phase on one hand, and increases its packet overhead and decreases its scalability on the other hand.

As AODV, ARAN is a reactive routing protocol that uses broadcasting in the route discovery process. However, ARAN uses cryptographic certificates to prevent most of the attacks against Ad-Hoc routing protocols and detect erratic behavior. The usage of these certificates increases the packet overhead, processing overhead, and route acquisition latency compared to AODV.

ARAN suffers from the centralized trust and load, i.e.; the single point of attack and failure. Moreover, it has a scalability problem due to using one certificate server (which may be the operation bottleneck), and the increased packet and processing overheads due to broadcasting the route request to the whole network along with the encryption/decryption processes.

ARANz is a secure restricted directional flooding routing protocol that adopts the authentication methods used with ARAN. Using restricted directional flooding to discover routes may increase the effect of a failure or movement of a single node. However, via dealing with the network as zones and using restricted directional flooding, our new model aims to exhibit better scalability and performance.

As opposed to ARAN, ARANz tries to distribute load and trust by dividing the area into zones and introducing multiple LCAs in each zone. This will help in achieving high level of security and robustness, and avoiding the single point of failure and attack problems. Using multiple LCAs in ARANz, on the other hand, comes up with a need to keep them synchronized.

It is obvious that ARANz is a scalable protocol since it will be able to maintain a good performance even with large networks. This is due to using restricted directional flooding instead of broadcasting, dividing the area into zones and distributing load among multiple LCAs.

6.0 CONCLUSION

A new model of routing protocol, ARANz, has been proposed in this work. This protocol addresses the managed-open environment where the possibility to use already established infrastructure is available. ARANz introduces a hierarchical and distributed routing algorithm, which improves performance and scalability of the routing protocol by dividing the area into zones. ARANz aims to achieve robustness, increases network security and solves the single point of failure and attack problems by introducing multiple LCAs. ARANz also tries to exhibit better scalability, performance, and robustness against frequent topological changes via the restricted directional flooding position-based routing protocols. An overview and a qualitative comparison between AODV, ARAN and ARANz protocols have been presented in this paper.

7.0 FUTURE WORK

Due to the large number of nodes and the large geographical area of Ad-Hoc networks a simulation tool will be used to evaluate and study the performance of the new protocol. Our next tasks are to evaluate the effectiveness of the protocol in dealing with security issues. Comparisons will then be performed with existing protocols. We also aim to test ARANz's scalability in relatively high node mobility environment, large area networks, and different number of malicious nodes.

ACKNOWLEDGEMENT

This work is done under the VotF University Malaya fund no. FS132/2008C.

REFERENCES

[1] L. K. Qabajeh, M. L. Mat Kiah, M. M. Qabajeh, 2009, "A Scalable, Distributed and Secure Routing Protocol for MANETs", *Proceedings of The International Conference on Computer Engineering and Applications (ICCEA 2009)*, pp. 51-56, Manila, Philippine.

[2] L. K. Qabajeh, M. L. Mat Kiah, M. M. Qabajeh, 2009, "A Scalable Secure Routing Protocol for MANETs" , *Proceedings of The International Conference on Computer Technology and Development (ICCTD 2009)*, pp.143-147, Kota Kinabalu, Malaysia.

[3] C. Perkins & E. Royer, 1999, "Ad hoc on-demand distance vector routing", *IEEE Proceedings of The 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100, New Orleans, LA.

[4] K. Sanzgiri, D. LaFlamme, B. Dahill, B. Levine, C. Shields & E. Belding-Royer, 2005, "Authenticated Routing for Ad Hoc Networks", *IEEE Journal On Selected Areas In Communications*, Vol. 23, No. 3, pp. 598 - 610.

[5] N. Beijar, 1998, "Zone Routing Protocol (ZRP)", Networking Laboratory, Helsinki University of Technology, Finland.
Available at: http://www.netlab.hut.fi/opetus/s38030/k02/Papers/08-Nicklas.pdf

[6] T. Lin, 2004, "Mobile Ad-hoc Network Routing Protocols: Methodologies and Applications", Ph.D. thesis, Faculty of the Virginia Polytechnic Institute and State University, Blacksburg, Virginia.
Available at: http://scholar.lib.vt.edu/theses/available/etd-03262004-144048/unrestricted/Tao_PhD_Dissertation.pdf

[7] M. Abolhasan, T. Wysocki & E. Dutkiewicz, 2004, "A review of routing protocols for mobile ad hoc networks", *Ad Hoc Networks*, Vol. 2, No. 1, pp. 1-22, Elsevier.

[8] Y. Cao & S. Xie, 2005, "A Position Based Beaconless Routing Algorithm for Mobile Ad hoc Networks", *Proceedings of The International Conference on Communications*, Circuits and Systems, Vol. 1, pp. 303- 307, IEEE.

[9] H. Li & M. Singhal, 2006, "A Secure Routing Protocol for Wireless Ad Hoc Networks", *Proceedings of The 39th Annual Hawaii International Conference on System Sciences (HICSS '06)*, Vol. 9, pp. 225a - 225a, IEEE.

[10] Sk. Mizanur Rahman, M. Mambo, A. Inomata & E. Okamoto, 2006, "An Anonymous On-Demand Position-based Routing in Mobile Ad Hoc Networks", *Proceedings of The International Symposium on Applications and the Internet*, pp. 300-306, Arizona, USA.

[11] G. Zapata, 2002, "Secure Ad hoc On-Demand Distance Vector Routing", *ACM Mobile Computing and Communications Review (MC2R)*, Vol. 6, No. 3, pp. 106–107.

[12] Y. Hu, A. Perrig & D. Johnson, 2002, "ARIADNE: A Secure On-Demand Routing Protocol for Ad Hoc Networks", *Proceedings of The ACM International Conference on Mobile Computing and Networking (MOBICOM'02)*, pp. 12–23, Georgia, USA.

[13] V. Giruka & M. Singhal, 2005, "Angular Routing Protocol for Mobile Ad-hoc Networks", *Proceedings of The 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)*, pp. 551-557.

[14] B. Karp & H. Kung, 2000, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", *Proceedings of The 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 2000)*, pp. 243-254, Massachusetts, USA.

[15] X. Wu, 2005, "VPDS: Virtual Home Region based Distributed Position Service in Mobile Ad Hoc Networks", *Proceedings of The 25th IEEE International Conference on Distributed Computing Systems (ICSCS 2005)*, pp. 113-122.

[16] Y. Ko & N. Vaidya, 2000, "Location-Aided Routing (LAR) in mobile ad hoc networks", *Wireless Network (WINET)*, Vol. 6, No. 4, pp. 307-321, ACM.

[17] L. Blazevic, L. Buttyan, S. Capkum, S. Giordano, J. Hubaux & J. Le Boudec, 2001, "Self-organization in mobile ad-hoc networks: the approach of terminodes", *IEEE Communication Magazine*, Vol. 39, No. 6, pp. 166-174.

[18]S. Carter & A. Yasinsac, 2002, "Secure Position Aided Ad Hoc Routing", *Proceedings of The IASTED International Conference on Communications and Computer Networks (CCN02)*, pp. 329-334, Cambridge.

[19] J. Song, V. Wong & V. Leung 2007, "Secure position-based routing protocol for mobile ad hoc networks", *Elsevier Ad Hoc Networks Journal*, Vol. 5, No. 1, pp. 76–86, Elsevier.

[20] F. Sivrikaya & B. Yener, 2004, "Time Synchronization in Sensor Networks: A Survey", *IEEE Network*, Vol. 18, No. 4, pp. 45-50.

BIOGRAPHY

Liana Khamis Qabajeh received her B.Sc. from Palestine Polytechnic University (PPU), Palestine in 2000 in Computer Engineering and joined the Engineering and Technology Faculty, PPU, as a research assistant. She received her M.Sc. from Jordan University of Science and Technology, Jordan in 2005 in Computer Engineering. Between 2005 and 2008 before pursuing her study, she was primarily involved in academic teaching and research in PPU. She is now working towards Ph.D. in Computer Science in University of Malaya, Malaysia. Her current research interests include Distributed Systems and Ad-Hoc Networks.

Dr. Miss Laiha Mat Kiah received her B.Sc. (Hons) in Computer Science from University of Malaya (UM), Malaysia in 1997, M.Sc. in 1998 and Ph.D. in 2007 from Royal Holloway, University of London UK. She joined Faculty of Computer Science & Information Technology, UM as a tutor in 1997. Between 1999 and 2003 before pursuing her study, she was primarily involved in academic teaching and research in the same faculty of UM. She was appointed as a lecturer in 2001 and as a senior lecturer in 2008. She was the Head of Computer Systems and Technology Department from Aug 2008 until Aug 2009. Her current research interests include key management, secure group communication and wireless mobile security. She is also interested in routing protocols and mobile Ad-Hoc networks.

Mohammad M. Qabajeh received his B.Sc. from Palestine Polytechnic University, Palestine in 2000 and M.Sc. from Jordan University of Science and Technology, Jordan in 2006 in computer Engineering. He worked as a network administrator in the periods (2000-2003) and (2006-2008). During these periods he worked as a part time lecturer in many Palestinian universities. He is now working towards Ph.D. in Computer Engineering in International Islamic University Malaysia, Malaysia. His current research interests include Distributed Systems and Ad-Hoc Networks.