SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.563

RESEARCH ARTICLE

A more secure and scalable routing protocol for mobile ad hoc networks

Liana Khamis Qabajeh[1]*, Miss Laiha Mat Kiah[1] and Mohammad Moustafa Qabajeh[2]

[1] Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia
[2] Department of Electrical and Computer Engineering, IIUM, Kuala Lumpur, Malaysia

ABSTRACT

An essential problem in mobile ad hoc networks is finding an efficient and secure route from a source to an intended destination. In this paper, we have proposed a new model of routing protocol named ARANz, which is an extension of the original Authenticated Routing for Ad Hoc Networks (ARAN). ARANz adopts the authentication methods used with ARAN and aims to increase security, achieve robustness and solve the single point of failure and attack problems by introducing multiple local certificate authority servers. Additionally, via dealing with the network as zones and using restricted directional flooding, our new model exhibits better scalability and performance. Through simulation, we evaluated ARANz and compared it with the original ARAN as well as Ad Hoc On-demand Distance Vector. Simulation results show that ARANz is able to effectively discover secure routes within relatively large networks with large number of mobile nodes, while maintaining the minimum packet routing load. Copyright © 2012 John Wiley & Sons, Ltd.

KEYWORDS
position based; secure; distributed; scalable; routing; routing protocol; mobile; ad hoc; ad hoc networks; wireless networks; location service

***Correspondence**
Liana Khamis Qabajeh, Computer Science and Information Technology Faculty, University of Malaya, Kuala Lumpur, Malaysia.
E-mail: liana_tamimi@ppu.edu

1. INTRODUCTION

Ad hoc wireless networks are self-organizing multi-hop wireless networks. Efficient routing is an important issue in ad hoc wireless networks because all nodes in the network act as hosts and routers. Furthermore, the concept and structure of ad hoc networks make them prone to be attacked using several techniques such as modification, impersonation and fabrication.

Considering ad hoc networks environments, managed-open environment is the one where most research is being carried out. Such type of ad hoc networks might be found among peers at a conference or students on a campus. In this type of environment, there is a possibility to use already established infrastructure. This means that there is an opportunity for pre-deployment or exchange of public keys, session keys or certificates. Although, without online trusted servers as in wired networks, it is difficult to be acquainted with the trustworthiness of each node, thus keeping malicious nodes away from the routes. However, the approach where one centralized server is used is impractical for ad hoc network. This server could be the operation bottleneck as it may be simply a normal ad hoc node with limited memory, CPU processing capacity and battery power. To tackle this problem, the certificate authority and position service system should be distributed among several servers deployed in the network.

The need for scalable and energy efficient protocols, together with the recent availability of small, inexpensive and low-power positioning instruments justify introducing position-based routing 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 hierarchal and distributed routing protocol called ARANz has been proposed in this work. Based on the original Authenticated Routing for Ad Hoc Networks (ARAN) [1], ARANz aims to improve performance of the routing protocol and distribute routing load by dividing the area into zones. Moreover, it seeks 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 certificate authority servers. Finally, ARANz aspires to exhibit better scalability, performance and robustness against frequent topological changes by utilizing the idea of restricted directional flooding. Subsequently, in conjunction with the chosen routing strategy, a distributed location service has been proposed.

This paper is an extension of our work in [2]. A qualitative comparison among Ad Hoc On-demand Distance Vector (AODV) [3], ARAN and ARANz protocols have been presented in [2]. This paper, on the contrary, provides detailed discussions of the new protocol ARANz, security analysis of both protocols (ARAN and ARANz), as well as simulated network performance evaluation and comparison among AODV, ARAN and ARANz protocols. From the results, we found that ARANz is able to discover secure routes effectively within relatively large networks with large number of nodes.

The rest of the paper is organized as follows. Section 2 introduces the existing and recent works on ad hoc routing protocols as well as introducing attacks against and security requirements of ad hoc networks. Section 3 presents our new routing protocol. Section 4 contains security analysis as well as a simulated comparison among AODV, ARAN and ARANz protocols. We analyze and discuss our findings in Section 5 and conclude our work in Section 6. Finally, we present our future direction in Section 7.

2. BACKGROUND

We start this section with discussions on the existing works of ad hoc routing protocols. Then subsections 2.2 and 2.3 present security requirements, as well as attacks targeted against ad hoc networks.

2.1. Existing routing protocols

In general, ad hoc networks routing protocols are divided into two main categories: *topology based* and *position based*. *Topology-based* routing protocols use information about links existing in the network to perform packet forwarding. They are, in turn, divided into three categories: *proactive*, *reactive* and *hybrid* protocols. *Proactive* routing protocols periodically broadcast control messages aiming to have each node always know a current route to all destinations. Proactive routing protocols are less suitable for ad hoc wireless networks because they constantly consume power throughout the network, regardless of the network activity, and they are not designed to track topology changes occurring at a high rate [4,5]. On the contrary, *reactive* routing protocols are deemed more appropriate for wireless environments because they initiate a route discovery process only if a source has data to be sent to a specific destination. One advantage of reactive routing protocols, such as AODV, is that no periodic routing packets are required. However, they may have high control overhead in networks with high mobility and heavy traffic loads. Also, they suffer from a scalability problem because of blind broadcasts carried out for route discovery [5]. Zone Routing Protocol (ZRP) [4] is an example of *hybrid* routing protocols that aims to combine the best properties of both proactive and reactive approaches. The disadvantage of ZRP is that for large routing zones, the protocol can behave like a pure proactive protocol, whereas for small zones, it behaves like a reactive protocol [6].

In general, topology-based protocols are not scalable for networks with more than several hundred of nodes [7]. Additionally, none of ad hoc routing protocols mentioned earlier defines their security requirements, and they inherently trust all participants. Obviously, this could result in security vulnerabilities and exposures that could easily allow routing attacks [1,8,9]. After that, many secure routing protocols were proposed such as [1,10–12]. One protocol of interest is the ARAN protocol. ARAN is similar to AODV but provides authentication of route discovery, setup and maintenance as well as message integrity and non-repudiation. The main objective of ARAN is to protect against attacks from malicious nodes in a managed-open environment where a small amount of prior security coordination is expected. ARAN requires the existence of a trusted certificate authority (CA) server. In comparison with basic AODV, ARAN prevents a number of attacks such as modification, impersonation and fabrication exploits. On the contrary, ARAN causes more packet overhead and higher route discovery latency because each packet must be signed. Besides, it has problems handling scalability with the number of nodes. ARAN also based on a centralized trust hence suffers from the compromised server problem and single point of failure.

In recent developments, *position-based* routing protocols exhibit better scalability, performance and robustness against frequent topological changes [7]. 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 via Global Positioning System (GPS) and geographical position of the destination using a location service. There are different approaches for position-based routing protocols that are categorized into three main groups: *greedy*, *restricted directional flooding* and *hierarchical* routing protocols.

In *greedy* forwarding, such as *Greedy Perimeter Stateless Routing* [13], each intermediate node selects a neighboring node that is closest to the destination as the next hop. Hence, nodes periodically broadcast small beacons to announce their position to their neighbors. Periodic beaconing consumes nodes energy and network bandwidth [7]. Also, greedy forwarding may not always find the optimum route especially in sparse networks [13,14]. In *restricted directional flooding*, such as *Location-Aided Routing* [15], the sender broadcasts the packet to all single hop neighbors towards the destination. Upon receiving a route request message, the receiver node retransmits the message if it is closer to the destination than its previous hop; otherwise, the message is dropped. *TERMINODES* [16] is an example of *hierarchical* routing protocols in which packets are routed based on a proactive distance vector if the destination is close to the sender and greedy forwarding is used in long distance routing.

All the aforementioned position-based protocols are susceptible to various security attacks because they did not consider security issues [9]. Recently, a few secure position-based routing protocols have been proposed for ad hoc networks. Examples of these are *Secure Position Aided Ad Hoc Routing* [17], *Anonymous On-Demand Position-based Routing in Mobile Ad Hoc Networks* [9] and *Secure Geographic Forwarding* [18]. However, they suffer from some problems such as single point of failure and attack, increased packet and processing overhead and/or scalability problems.

From observations, we note that many topology-based routing protocols still have 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. Ad hoc networks security requirements

To ensure the security of ad hoc network, a number of requirements need to be satisfied. These requirements are summarized as follows [19–21]:

- Availability: the network should remain operational and available to send and receive messages at any time. Availability ensures the survivability of the network even if the network is under attack.
- Confidentiality: provides secrecy to sensitive data being sent over the network; the contents of every message can be understood only by its source and destination.
- Integrity: ensures that messages being sent over the network are not corrupted by intentional or accidental modification.
- Authentication: ensures the identity of nodes in the network; assure that they are who they claim to be.
- Non-repudiation: guarantees that neither sender nor receiver can deny that he has sent or received the message.

Recent development also indicated that privacy has become an important security issue, and plenty of works about anonymous ad hoc routing protocol have been researched such as [9,22–24]. The *anonymity* in an ad hoc routing means that the identity of node, route path information and location information must be veiled from not only an adversary but also from other nodes.

2.3. Attacks against ad hoc networks routing protocols

Ad hoc routing is a very fundamental operation on an ad hoc network; hence, it has been a main target for an attacker to disrupt an ad hoc network [25]. Two kinds of attacks that can be launched against ad hoc networks are [20,26,27]:

- Passive attacks: the attacker does not disturb the routing protocol. He only eavesdrops on routing traffic and endeavors to extract any valuable information such as node hierarchy and network topology.
- Active attacks: in active attacks, malicious nodes can disturb the correct functioning of a routing protocol by modifying routing information, fabricating false routing information and/or impersonating other nodes. The following is a brief explanation of these active attacks:
- *Modification attacks* are normally targeted against the integrity of routing computation. By modifying routing information, an attacker can cause network traffic to be dropped, redirected to a different destination or to take an extended route to the destination.
- *Fabrication attacks* are performed by generating deceptive routing messages. These attacks are difficult to identify as they are received as legitimate routing packets.
- During *impersonation attacks*, a malicious node can launch many attacks and misrepresent the network topology by masquerading as another legitimate node through spoofing.

3. PROPOSED PROTOCOL

In this section, we propose a new routing model called ARANz. The proposed protocol is named ARANz because it adopts the authentication steps used with the ARAN protocol and deals with the network as zones. The following subsections discuss the proposed protocol considering different phases it is composed of.

3.1. Introduction and important assumptions

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. If a misbehavior detection scheme is present on the network, then the security of our protocol can be improved through collaboration with this scheme. Moreover, ARANz tries to exhibit better scalability, performance and robustness against frequent topological changes by utilizing the idea of restricted directional flooding. Hence, LCAs work also as position servers, and each node should inform LCAs of its zone about its new position if it has moved.

ARANz consists mainly of five phases: network setup, network maintenance, location service, route instantiation and maintenance and, finally, data transmission. *Network setup phase* 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 instantiation and maintenance phase* is initiated by sending a route discovery packet (RDP) using restricted directional flooding. After finishing route discovery and setup, the source begins *data transmission phase* by sending the data to the destination.

We assume (Nn) cooperative nodes in a managed-open environment. These nodes are distributed randomly in a square-shaped area and are aware of their positions. A particular node in the network is chosen to have the software needed to initiate the network setup, divide the area into zones and elect the initial LCAs. This node is called the primary certificate authority (PCA) server and possesses the private part of the network key (K_{NET-}). PCA is chosen prior to the network deployment; this is possible because we are dealing with managed-open environment. 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) that is used for encryption and decryption of the packets sent by all 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.

Before proceeding further, let us define the following variables, notations and packet identifiers that will be used in the upcoming sections. Table I shows variables and notations used for ARANz protocol, whereas Table II presents the notation used in presenting the proposed LCAs election algorithm. Table III summarizes the used packet identifiers.

3.2. Network setup

The PCA starts the network setup by broadcasting a NETwork SETup (NETSET) packet notifying the nodes about the beginning of this phase. 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. Suppose that node *P* has been chosen to play the role of the PCA, it will broadcast the following NETSET packet:

$$\text{PCA} \rightarrow \text{broadcast} : [\text{NETSET},\ \text{IP}_\text{P}]\ \ \text{K}_{\text{NET}-}$$

Each node, upon receiving the first NETSET packet, records the IP address of the previous node, continues broadcasting the packet and replies with a Node INformation (NIN) packet to the PCA. NIN packet contains the node's IP address (IP_A), the node it originally received the NETSET packet from (IP_D) along with the needed information to elect the LCAs such as position (P_A), speed (S_A), battery remaining lifetime (B_A), CPU power (C_A) and memory (M_A). For example, node *A* will send the following packet:

$$\text{A} \rightarrow \text{PCA} : (\text{NIN}, \text{IP}_\text{A}, \text{IP}_\text{D}, [\text{P}_\text{A}, \text{S}_\text{A}, \text{B}_\text{A}, \text{C}_\text{A}, \text{M}_\text{A}]\, \text{K}_{\text{A}-})\ \text{CK}$$

Information about each node is signed using the node's private key (K_{A-}) to enable PCA to ensure that the node that sent the packet is truly the node claiming that and to ensure nodes' privacy by assuring that the PCA is the sole node that reads this private information. Moreover, the NIN packets are encrypted using the CK. Hence, each node upon the receipt of a NIN packet tries 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. The NIN packet is sent to PCA through the reverse path; that is, each node forwards the NIN packet to the node it originally received the first NETSET packet from.

After receiving the NIN packets from all authorized nodes existing currently in the network, PCA will divide

Table I. Variables and notations for ARANz.

Notation	Description	Notation	Description
PCA	Primary certificate authority server	LCA_{zs}	Local certificate authority *s* of zone *z*
CK	Common key	Nn	Number of nodes
IP_n	IP address of node *n*	P_n	Position of node *n*
K_{n-}	Private key of node *n*	K_{n+}	Public key of node *n*
K_{NET-}	Private key of the network	K_{NET+}	Public key of the network
K_{Zz-}	Private key of zone *z*	K_{Zz+}	Public key of zone *z*
$[d]K_{n-}$	Data *d* digitally signed by node *n*	$\{d\}K_{n+}$	Data *d* encrypted with key K_{n+}
$CertLZ_z$	Zone *z* LCAs certificate	$Cert_n$	Node *n* certificate
t	Time stamp	*e*	Certificate expiration time
N_n	Nonce issued by node *n*	N_{Zz}	Nonce issued by zone *z*
$Coord_{Zz}$	Coordinates of zone *z*	D_{LCAzs}	Distance to LCA_{zs}
$LCAsZ_z$	Identities and positions of LCAs in zone *z*	$8NbrZ_Z$	Eight-neighboring zones of zone *z* and their coordinates

Table II. Variables and notations for the proposed LCAs election algorithm.

Notation	Description	Notation	Description
ProbL_{nzs}	Probability of node *n* existing in zone *z* to be elected as a LCA of side *s*	D_{nzs}	Distance between position of node *n* existing in zone *z* and the middle point of the zone side *s*
S_n	Speed of node *n*	B_n	Battery lifetime of node *n*
C_n	CPU power of node *n*	M_n	Memory of node *n*
Dmax	Maximum possible distance between a node and middle point of a zone side	Wd	Weight of distance between a node and middle point of a zone side
Smax	Maximum possible node movement speed	Ws	Weight of node movement speed
Bmax	Maximum possible node battery lifetime	Wb	Weight of node battery remaining life
Cmax	Maximum node CPU power	Wc	Weight of node CPU power
Mmax	Maximum node memory capacity	Wm	Weight of node memory capacity
x_n	*x*-Coordinate of node *n*	y_n	*y*-Coordinate of node *n*
xc_{zc}	*x*-Coordinate of corner *c* of zone *z*	yc_{zc}	*y*-Coordinate of corner *c* of zone *z*
xm_{zs}	*x*-Coordinate of middle point of side *s* of zone *z*	yc_{zs}	*y*-Coordinate of middle point of side *s* of zone *z*

Table III. Packet identifiers for ARANz.

Packet identifier	Stand for	Packet identifier	Stand for
NETSET	NETwork SETup	NIN	Node INformation
NROLE	Node ROLE	CREQ	Certificate REQuest
ACREQ	Acceptance of Certificate REQuest	ACREP	Acceptance for Certificate REPly
CREP	Certificate REPly	NCERT	Node CERTificate
DNODE	Departed NODE	NNODE	New NODE
NZONE	New ZONE	NLCAP	New LCA Position
NALCAP	New Adjacent LCA Position	NLCAE	New LCA Election
NPROB	Node PROBability	FLCA	Failed LCA
FALCA	Failed Adjacent LCA	NLCA	New LCA
NALCA	New Adjacent LCA	FNODE	Failed NODE
EZONE	Empty ZONE	MNODE	Misbehavior NODE
CNODE	Compromised NODE	PDP	Position Discovery Packet
PREP	Position REPly	RDP	Route Discovery Packet
RREP	Route REPly	ERR	ERRor

the network area into multiple equal-sized virtual zones with multiple initial LCAs assigned for each zone. In our implementation environment, we consider to use square-shaped zones with four LCAs. We note that this choice may seem a bit rigid for now, but we believe that this will be the starting point for future work in regard to the implementation of other zone shapes and dynamic number of LCAs.

As a subsequent step, PCA informs nodes of the initial role that each node will play: LCA or regular node. The used algorithm to elect the initial LCAs is explained in subsection 3.2.1, whereas subsection 3.2.2 tackles notifying the nodes about the initial roles that they will play. The network structure is shown in Figure 1, if we suppose that the entire area is divided, for example, into four zones.

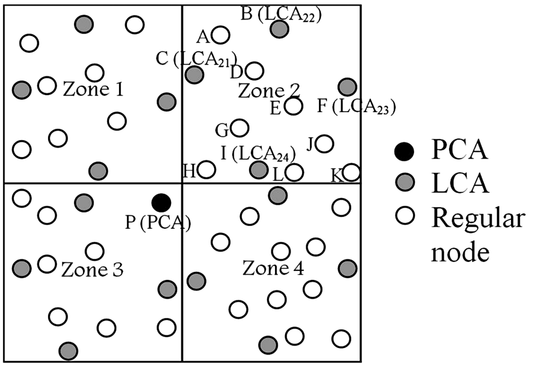


Figure 1. Network structure after electing initial LCAs.

3.2.1. LCAs election

After dividing area into zones, PCA begins the process of electing LCAs. Upon electing the initial LCAs, each node *n* inside a specific zone *z* is assigned a weight representing its probability of being a LCA to a particular zone side *s*. The most important points in selecting LCAs are distance between the node and the middle point of the zone side that the LCA will be responsible for (D_{nzs}), node's speed (S_n) and battery remaining lifetime (B_n). Choosing a LCA that is close to the middle point of the zone side and moving with a low speed increases the probability that the communication between LCAs of different zones will be carried out using one hop, which helps in protecting these important packets. Choosing LCAs with low movement speed also increases the probability that the elected LCA will stay longer in the zone, and so no need to re-elect a new LCA within a short period. Moreover, choosing a node with high battery remaining lifetime reduces the probability of having its battery energy drained. Another two important factors that must be considered when electing a LCA are node's CPU processing power (C_n) and memory (M_n). LCAs with high CPU processing power and large memory significantly affect network performance because these LCAs could be the operation bottleneck for the position management scheme.

Primary certificate authority uses NIN packets that it receives to calculate the probability of each node to be elected as a LCA. Probability of node n existing currently in zone z to be elected as a LCA of a particular side s is given as:

$$\begin{aligned}\mathrm{ProbL}_{nzs} = {} & \mathrm{Wd}_{\times}(1-(\mathrm{D}_{nzs}/\mathrm{Dmax})) \\ & +\mathrm{Ws}_{\times}(1-(\mathrm{S}_{n}/\mathrm{Smax}))+\mathrm{Wb}_{\times}(\mathrm{B}_{n}/\mathrm{Bmax}) \\ & +\mathrm{Wc}_{\times}(\mathrm{C}_{n}/\mathrm{Cmax})+\mathrm{Wm}_{\times}(\mathrm{M}_{n}/\mathrm{Mmax})\end{aligned}$$

Where

Wd, Ws, Wb, Wc and Wm: weighting factors
Dmax: maximum possible distance between a node and middle point of a zone side
Smax: maximum possible node movement speed
Bmax: maximum possible battery lifetime
Cmax: maximum CPU power found in the market
Mmax: maximum memory capacity exists in the market

Distance D_{nzs} between node's position $P_n = (x_n, y_n)$ and middle point (xm_{zs}, ym_{zs}) of side s in zone z is given as follows:

$$\mathrm{D}_{nzs} = \left((x_n - \mathrm{xm}_{zs})^2 + (y_n - \mathrm{ym}_{zs})^2\right)^{1/2}$$

Dmax is calculated once. Referring to Figure 2, Dmax can be calculated, for example, as the distance between the middle point (xm_{13}, ym_{13}) of side 3 in zone 1 and one of the zone corners in front of it ((xc_{11}, yc_{11}) or (xc_{14}, yc_{14})).

$$\mathrm{Dmax} = \left((x_{\mathrm{c11}} - x_{\mathrm{m13}})^2 + (y_{\mathrm{c11}} - y_{\mathrm{m13}})^2\right)^{1/2}$$

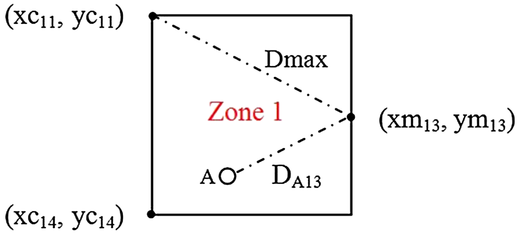

$(\mathrm{xc}_{11}, \mathrm{yc}_{11})$
Dmax
Zone 1
$(\mathrm{xm}_{13}, \mathrm{ym}_{13})$
A
$\mathrm{D}_{\mathrm{A13}}$
$(\mathrm{xc}_{14}, \mathrm{yc}_{14})$

Figure 2. Maximum possible distance between middle point of a zone side and a node inside that zone.

Smax is a pre-defined value that depends on the environment where the protocol is deployed. Bmax, Cmax and Mmax are pre-defined values that depend on the current technology found in the market.

The elected LCAs (from initial setup) can move freely in the network (including move in and out of the network), as well as becoming corrupted, destroyed or compromised. These issues are considered upon developing our protocol and are discussed in subsection 3.3.

3.2.2. Roles notification

After dividing the area into zones and electing the LCAs, PCA will unicast a Node ROLE message (NROLE) to each participant node. Source routing is used to send these messages because the PCA knows the position of all nodes in the network. These messages enable each node to know its role in the network: LCA or non-LCA (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 time stamp (t) of when the certificate was created and a time (e) at which the certificate expires. These variables are concatenated and signed with the $\mathrm{K}_{\mathrm{NET}-}$. Nodes use these certificates to authenticate themselves to other nodes during the exchange of network maintenance, position and routing packets. PCA will send the following NROLE message to node E, for example:

$$\begin{aligned}\mathrm{PCA} \rightarrow \mathrm{E} : & [\mathrm{NROLE},\ \mathrm{IP_P},\ (\mathrm{IP_H},\ \mathrm{IP_G},\ \mathrm{IP_E}), \\ & (\text{“R”},\ \mathrm{Cert_E},\ 2,\ \mathrm{LCAsZ_2},\ \mathrm{K_{Z2+}})\ \mathrm{K_{E+}})\ \mathrm{K_{NET-}}\end{aligned}$$

Where

($\mathrm{IP_H}$, $\mathrm{IP_G}$, $\mathrm{IP_E}$): is the source route
"R": indicates that E is a regular node
$\mathrm{Cert_E}$: node's E certificate; $\mathrm{Cert_E} = [\mathrm{IP_E}, \mathrm{K_{E+}}, t, e]\ \mathrm{K_{NET-}}$
2: is the zone number of node E
$\mathrm{LCAsZ_2}$: ($\mathrm{IP_{LCA21}}$, $\mathrm{P_{LCA21}}$, $\mathrm{IP_{LCA22}}$, $\mathrm{P_{LCA22}}$, $\mathrm{IP_{LCA23}}$, $\mathrm{P_{LCA23}}$, $\mathrm{IP_{LCA24}}$, $\mathrm{P_{LCA24}}$)
$\mathrm{K_{Z2+}}$: zone number 2 public key

These NROLE messages are signed using the $\mathrm{K}_{\mathrm{NET}-}$ to ensure that the PCA is the source of the message. Also, the private information is encrypted using the node's public key ($\mathrm{K_{E+}}$) to ensure that the corresponding node is the only node that is able to decrypt this critical and important information. Each node along the path forwards the packet to its next hop in the source route.

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 eight-neighboring zones ($\mathrm{8NbrZ_Z}$), 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 an authentication table. Moreover, it contains an absent nodes list containing IP addresses and public keys of authorized nodes that were not in the network during network setup. This list enables absent nodes to join the network from any zone at any time. Node *I* for example will receive the following NROLE message:

$$PCA \rightarrow I : [NROLE,\ (IP_H,\ IP_I), (\text{“L”},\ Cert_I,\ CertLZ_3,\ 4, 2, Coord_{Z2},\ 8NbrZ_2,\ K_{Z2-},\ K_{Z2+},\ LCAsZ_2,\ IP_{LCA42},\ P_{LCA42}, K_{Z4+}, part_{of}K_{Z4-}, \text{authentication table, absent nodes}) K_{I+}) K_{NET-}$$

Where

(IP_H, IP_I): the source route
"L": indicates that *I* is a LCA
$CertLZ_2$: zone number 2 LCAs certificate; $CertLZ_2 = [2, K_{Z2+}, t, e]\ K_{Z2-}$
4: number of that LCA in its zone
2: the zone number of *I*
$Coord_{Z2}$: coordinates of zone number 2
K_{Z2-}, K_{Z2+}: private/public key pair of zone number 2
$8NbrZ_2$: (1, $Coordinate_{Z1}$, 3, $Coordinate_{Z3}$, 4, $Coordinate_{Z4}$)
$LCAsZ_2$: (IP_{LCA21}, P_{LCA21}, IP_{LCA22}, P_{LCA22}, IP_{LCA23}, P_{LCA23})
IP_{LCA42}, P_{LCA42}, K_{Z4+}, part_of_K_{Z4-}: LCA_{24} immediate neighboring zone information
Authentication table: ((IP_A, K_{A+}, t, e, P_A), (IP_B, K_{B+}, t, e, P_B), . . .)
Absent nodes: (IP_U, K_{U+}, IP_Z, K_{Z+})

The authentication table is used to update nodes' certificates and contains a tuple (IP address, public key, time stamp (t), certificate expiration time (e) and position) for each node inside this zone. The zone LCA certificate ($CertLZ_z$) binds zone's number to its public key and contains the zone number, zone public key, time stamp and certificate expiration time. 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. Table IV summarizes certificates' types used with our protocol, whereas Table V summarizes the different keys used.

3.3. Network maintenance

During the network lifetime, nodes can update their certificates, move freely in the network, move in and out the network in addition to becoming corrupted or destroyed. Our protocol tries to cope with these issues. By the end of the network setup phase, each node has its node certificate; these certificates are used to apply 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 a LCA, then it will also attach 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 (whether LCA

Table IV. Certificates' types used with ARANz.

Certificate	Issued to	Used for	Case
Node certificate ($Cert_n$)	All nodes	Nodes authentication	During the exchange of network maintenance, position and routing packets
Zone LCAs certificate ($CertLZ_z$)	LCAs	LCAs verification	During the exchange of network maintenance and position packets

Table V. Different keys used with ARANz.

Key	Used for
Common key (CK)	Encryption and decryption of packets sent by non-PCA nodes during network setup phase
Nodes private/public key pairs (K_{A-}/K_{A+})	• Encryption and decryption of packets sent by a specific node after network setup phase • Destination's public key may be used for encrypting datapackets to ensure data privacy
Network private/public key pairs (K_{NET-}/K_{NET+})	• Encryption and decryption of packets sent by PCA in network setup phase • Encryption and decryption of nodes' certificates
Zone private/public key pairs (K_{Z1-}/K_{Z1+})	Encryption and decryption of a particular zone LCAs certificate

or regular) validates the previous node's signature (using the previous node's public key, which is extracted from its certificate), removes previous node's certificate and signature, signs the original contents of the packet and appends its own certificate. Upon sending packets between adjacent LCAs or from LCAs to nodes in their zones, the destination node ensures that the source LCA has a fresh certificate for its zone by decrypting the attached zone LCA certificate using the public key of the source zone.

Another important point to be mentioned is that the packets sent from the regular nodes to LCAs of their zones are sent using restricted directional flooding because each node within that zone knows the position of these LCAs. Also, communications to establish routes between nodes (after obtaining the destination position) are carried out using restricted directional flooding. 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 because these LCAs are aware of positions of all 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 network.

Table VI shows the strategy used for sending different packets according to the source and destination of the packet. Restricted directional flooding and source routing are discussed in details in subsection 3.3.1; however, after that, we will only refer to the strategy of sending a particular packet.

3.3.1. Certifications update

All nodes in a specific zone must keep valid certificates with their zone's LCAs. This is achieved by periodically sending a Certificate REQuest (CREQ) packet to the nearest LCA to itself to keep overhead minimal. This CREQ packet is signed by node's private key and sent using restricted directional flooding. Referring to Figure 3, node K for example will send the following:

$$K \rightarrow \text{broadcast} : [\text{CREQ},\ \text{IP}_{\text{LCA34}},\ \text{N}_K]\ \text{K}_{K-},\ \text{Cert}_K$$

The CREQ packet includes a packet type identifier (CREQ), the IP address of the nearest LCA (IP_{LCA34}) and the node Nonce (N_K). The node's certificate is appended to the packet after signing it to enable nodes to validate the signature and verify that K's certificate has not expired. The purpose of the nonce is to uniquely identify a CREQ initiated by a particular source. Each time K performs certificate request, it monotonically increases this nonce. Hence, a given (IP_i, N_i) pair is used to check whether this CREQ is processed previously or not.

The first 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 forward back to the source. The receiving node uses K's public key, which it extracts from K's certificate, to validate the signature and verify that K's certificate has not expired. The receiving node also checks the (IP_K, N_K) tuple to verify that it has not already processed this CREQ; nodes do not forward packets with already-seen tuples. The receiving node adds a new field D_{LCA34} indicating the distance from itself to LCA_{34}, signs the contents of the packet, appends its own certificate and broadcasts the packet to its neighbors. Let J be a neighbor that has received the CREQ broadcast from K, then, it subsequently rebroadcasts:

$$J \rightarrow \text{broadcast} : [[\text{CREQ},\ \text{IP}_{\text{LCA34}},\ \text{N}_K]\ \text{K}_{K-},\ \text{D}_{\text{LCA34}}]\ \text{K}_{J-},\ \text{Cert}_K,\ \text{Cert}_J$$

Upon receiving the CREQ, J's neighbor L validates the signatures for both K; the CREQ initiator and J; the neighbor it received the CREQ from, using the certificates in the CREQ. L now will compare the recorded distance D_{LCA34} with the distance between itself and LCA_{34} (no need to include P_{LCA34} in the packet because it is known by all nodes in zone 3). If it is closer to LCA_{34}, then it will continue broadcasting the packet after changing the distance value in the packet, else the packet will be dropped. If the node L decided to rebroadcast the packet, then it removes J's certificate and signature, records J as its predecessor, signs the contents of the packet originally broadcast by K and appends its own certificate. Then, L will rebroadcast the CREQ:

Table VI. Strategies used for sending different packets in ARANz.

Packet sent		Packet-sending strategy
From	To	
Node	LCA in the same zone	Restricted directional flooding
Node	Node	Restricted directional flooding
LCA	LCA in a different zone	If destination is within the transmission range of source 1-hop unicast is used; else restricted directional flooding is used
LCA	LCA or Node in the same zone	Source routing

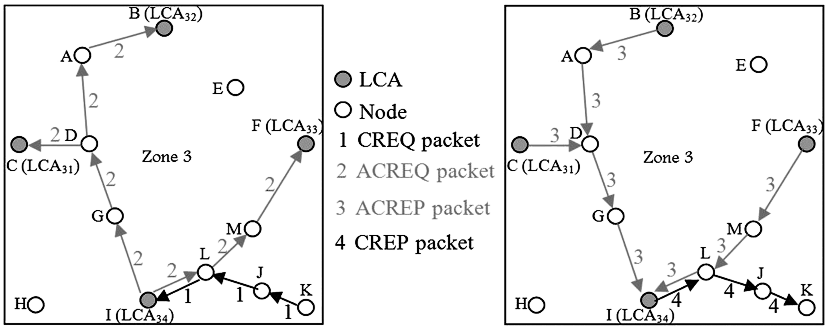


Figure 3. Node *K* certification update.

$$L \rightarrow \text{broadcast} : [[\text{CREQ},\ \text{IP}_{\text{LCA34}},\ \text{N}_{\text{K}}]\ \text{K}_{\text{K}-},\ \text{D}_{\text{LCA34}}]\ \text{K}_{\text{L}-},\ \text{Cert}_{\text{K}},\ \text{Cert}_{\text{L}}$$

Each intermediate node along the path repeats the same steps as *L* till reaching LCA_{34}, which replies to the first CREQ that it receives for a source and a particular certificate nonce. Each LCA upon receiving CREQ packet communicates other LCAs in its zone to ask them whether to update this certificate or not. This is carried out by sending a packet to each LCA asking for Acceptance of the Certificate REQuest (ACREQ). For example, LCA_{34} will send the following packet to LCA_{32} using source routing:

$$\text{LCA}_{34} \rightarrow G : [\text{ACREQ},\ (\text{IP}_{\text{D}},\ \text{IP}_{\text{A}},\ \text{IP}_{\text{LCA32}}),\ \text{CertLZ}_3,\ \text{IP}_{\text{K}},\ \text{N}_{\text{K}})\ \text{K}_{\text{LCA34}-},\ \text{Cert}_{\text{LCA34}}$$

where (IP_{D}, IP_{A}, IP_{LCA32}) is the source routing. So, each intermediate node in the path will check its next hop in the source route and send the packet to it, after validating the signatures for both the ACREQ initiator and the neighbor it received the ACREQ from, removing previous hop certificate and signature, recording its predecessor, signing the packet and appending its own certificate. CertLZ_3 is used by node LCA_{32} to ensure that the LCA that sent the ACREQ is really a LCA for its zone and has a fresh zone LCA certificate. Figure 3(a) shows sending the needed certificate request packets (CREQ and ACREQ) for updating node *K* certification.

So LCA_{34} will be allowed to issue a certificate only if it received Acceptance for Certificate REPly (ACREP) packet from most LCAs of that zone (signed by their private keys). This will help in increasing the robustness and security of the protocol; if one server failed or is compromised, then the other three servers will still be able to issue valid certificates (will be discussed in subsections 3.3.3 and 3.3.5). For example, LCA_{32} will send through the reverse path the following ACREP packet:

$$\text{LCA}_{32} \rightarrow A : [\text{ACREP},\ \text{IP}_{\text{LCA34}},\ \text{CertLZ}_3,\ \text{IP}_K,\ \text{N}_K]\ \text{K}_{\text{LCA32}-},\ \text{Cert}_{\text{LCA32}}$$

In the case that there are no failed or compromised servers discovered, LCA_{34} will be allowed to issue a fresh certificate for *K* after receiving at least three ACREP packets (one of them may be from itself). Then, LCA_{34} will unicast a Certificate REPly (CREP) packet back along the reverse path to the source. Let the first node that receives the CREP sent by LCA_{34} be node *L*:

$$\text{LCA}_{34} \rightarrow L : [\text{CREP},\ \text{CertLZ}_3,\ \text{IP}_K,\ \text{N}_K,\ \text{Cert}_K]\ \text{K}_{\text{LCA34}-},\ \text{Cert}_{\text{LCA34}}$$

This CREP includes a packet type identifier (CREP), the zone certificate that LCA_{34} has ($\text{Cert}_{\text{LCAZ3}}$), the IP address of *K* (IP_K), the certificate issued to *K* (Cert_K) and the nonce sent by *K*. $\text{Cert}_{\text{LCAZ3}}$ is used by node *K* to make sure that the LCA that issued the certificate is really a LCA for its zone and has a fresh zone LCA certificate.

Nodes that receive the CREP forward the packet back to the predecessor from which they received the original CREQ. Each node along the reverse path back to the source signs the CREP and appends its own certificate before forwarding the CREP to the next hop. Let *L*'s next hop to the source be node *J*:

$$L \rightarrow J : [[\text{CREP},\ \text{CertLZ}_3,\ \text{IP}_K,\ \text{N}_K,\ \text{Cert}_K]\ \text{K}_{\text{LCA34}-}]\ \text{K}_{L-},\ \text{Cert}_{\text{LCA34}},\ \text{Cert}_L$$

Node *J* validates *L*'s signature on the received packet, removes the signature and certificate, then signs the contents of the packet and appends its own certificate before unicasting the CREP to *K*:

$$J \rightarrow K : [[\text{CREP},\ \text{CertLZ}_3,\ \text{IP}_K,\ \text{N}_K,\ \text{Cert}_K]\ \text{K}_{\text{LCA34}-}]\ \text{K}_{J-},\ \text{Cert}_{\text{LCA34}},\ \text{Cert}_J$$

Figure 3 shows sending the needed certificate reply packets (ACREP and CREP) through the reverse paths. LCAs inside a specific zone carry identical information about nodes inside the zone to avoid single point failure. Each LCA upon issuing a certificate should unicast a Node CERTificate packet to other LCAs containing the newly issued certificate. For example, LCA_{34} will send (using source routing) the following packet to LCA_{32}:

$$LCA_{34} \rightarrow G : [NCERT,\ (IP_D,\ IP_{LCA32}),\ CertLZ_3,\ Cert_K]\ K_{LCA34-},\ Cert_{LCA34}$$

Local certificate authorities also must maintain fresh node and zone LCA certificates. Hence, periodically, each LCA should unicast ACREQ to other LCAs in its zone. And upon receiving the ACREPs, it will be issued both node and zone LCA certificates.

3.3.2. Nodes mobility

Once a regular node has moved a pre-defined distance 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 helps LCAs to keep track of up-to-date positions of nodes inside the zone and enables them to discover that a specific node has departed this zone to a neighboring one. If a node leaves to one of the immediate four-neighboring zones, then LCAs of the departed zone will remove 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. Figure 4 shows the communication done once node *R* left zone number 1 to zone number 2 assuming that the network is divided into four zones. 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 packets to other LCAs in its zone informing them about the new node.

In case that a node leaves to one of the diagonal D-neighboring zones (refer to Figure 4), LCA of the original zone sends a DNODE packet to adjacent LCA in the 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 a NZONE packet to the departing node. This LCA also will send multiple unicast packets to other LCAs in its zone telling them about the new node.

If any LCA has moved the pre-defined distance 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, a 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. In these two cases, a new LCA election is required. Upon deciding to leave its zone, LCA sends a New LCA Election (NLCAE) packet to nodes in its zone. Each node in the corresponding zone calculates its probability by itself to reduce the load on the leaving LCA. 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, the leaving LCA transfers the needed information about the zone to the new LCA.

3.3.3. Nodes failure

The sudden failure of a LCA (or node movement outside the network boundaries) can be discovered from the periodic 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, then 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. Hence, the failure of a single LCA (or even multiple LCAs) does not affect updating nodes' certificates as other LCAs in the zone collaborate to elect a new LCA.

After that, if the failed LCA has been repaired, then 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 are sent to all LCAs in the network. Each LCA in the corresponding zone sends 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.

Regular node failure also can be discovered from the periodic node certificate update. If a LCA had in its authentication table an expired node certificate and did not receive

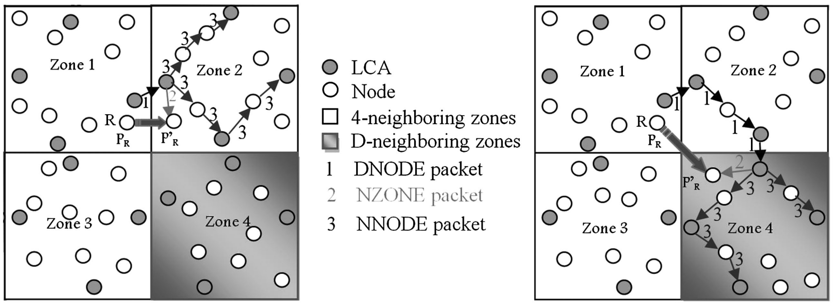


Figure 4. Movement of a node to a neighboring zone.

a CREQ packet within a pre-defined period, then 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.

3.3.4. Empty zones

Because of node movement, some zones may become empty. When many nodes leave a specific zone, the last four nodes that stay in that zone are its four LCAs. Any one of these LCAs that wants to leave the zone 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 to. This packet informs the LCA of the new zone that this node is the last node leaving the zone. This Empty ZONE packet will be sent to the eight-neighboring zones (four-neighboring zones and D-neighboring zones) of the empty zone informing them that this zone is empty.

Once a node left a specific zone and entered the empty zone, LCA of the departed zone sends a packet to the other immediate neighboring zones of the empty zone asking them to send the part of the empty zone private key they have. LCA of the departed zone, upon receiving these parts, combines them and sends a packet to the newly entered 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 till other nodes enter the zone. For example, if another node entered this zone, each one of them will play the role of two LCAs according to their positions.

3.3.5. Malicious and compromised nodes

Malicious nodes may cause some erratic behaviors such as the use of invalid certificates, improperly signed packets and misuse of some packets. ARANz drops all packets that have any erratic behavior. Our protocol can collaborate with a misbehavior detection system. If node *A* detects misbehavior of node *B*, node *A* excludes node *B* from future communications and sends Misbehavior NODE (MNODE) packet to report this to the nearest LCA in its zone. If most LCAs in a particular zone have received a pre-defined 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 till its certificate expires normally.

This technique is applicable also when the misbehaving node is a LCA. For example, if three LCAs of a particular zone received the pre-defined number of MNODE packets indicating misbehaving of the fourth LCA in their zone, they will remove this LCA from LCAsZ$_z$ list of this zone, broadcast a CNODE packet and initiate a new LCA election process. Even before revoking the certificate of the misbehaving LCA, the other three LCAs are still able to issue certificates for trusted nodes in their zone though the compromised LCA may refuse to send ACREP packets for the ACREQ packets it receives.

In the case of having two compromised LCAs at the same time in the same zone, neither the two trusted LCAs nor the compromised LCAs will be able to update certificates for the nodes in the zone. This situation may continue until the certificates of all nodes within the zone expire; in this state, they will not be able to participate in any future network activity. This situation also may end (before having nodes' certificates expired) by leaving one of the compromised LCAs to a neighboring zone or having its battery energy drained. In these cases, a new LCA needs to be elected to replace the compromised one. Having a third trusted LCA, the three trusted LCAs will be able to continue their tasks as usual.

On the contrary, this situation may end by replacing one of the well-behaved LCAs with a compromised one (e.g., the well-behaved LCA has moved to a neighboring zone and the newly elected one is a compromised node). This results in having three compromised LCAs in a particular zone at the same time. In this case, the security of the whole network is compromised, and these LCAs can collaborate together and issue certificates for untrusted nodes.

3.3.6. LCAs synchronization

All LCAs in the network should maintain synchronized clocks to ensure protocol correctness, to avoid a situation such that two nodes in different zones 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 adjusted to a reference clock in the network. Consequently, 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 he converted to the system's time. To maintain synchronization among different LCAs in the network, GPS or a synchronization scheme can be used. A simple synchronization scheme is proposed in the subsequent paragraphs.

As a starting point, PCA includes 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 newly elected LCA.

Additionally, all clocks are subject to clock drift; oscillator frequency will vary unpredictably because of various physical effects [28]. 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. To increase robustness of the system, the LCAs alternate this job. Also, a nonce is used to avoid replay attack. Certainly, the LCA includes its zone LCA 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 (subsection 3.5).

Regular nodes can use the time stamp included in their certificates 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.

3.4. Location service

This section discusses the location service used to enable any source to get the position of a specific destination. Two cases are considered: local and external communications. Local communications mean that the two communicating nodes reside inside the same zone. On the contrary, if a node has data to be sent to a node in another zone, then this is called external communication.

Before beginning the route discovery, the source should know the destination's position. The source *S* sends a position discovery packet (PDP) to the nearest LCA in its zone using restricted directional flooding to ask the LCA about the position of the destination *D*. Thus, source *S* in Figure 5 will send the following PDP to LCA_{24}:

$$S \rightarrow \text{broadcast} : [\text{PDP},\ N_S,\ IP_{LCA24},\ IP_D]\ K_{S-},\ Cert_S$$

The purpose of the node nonce N_S is to uniquely identify a PDP coming from a specific source. Each time *S* performs position request, it monotonically increases this nonce. The first node that receives this PDP adds a new field D_{LCA24} indicating the distance from itself to LCA_{24} to enable other nodes to continue restricted directional flooding.

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

$$LCA_{24} \rightarrow G : [\text{PREP},\ IP_S,\ N_S,\ P_D,\ CertLZ_2]\ K_{LCA24-},\ Cert_{LCA24}$$

If the destination is in a different zone, then the destination will not be found in the authentication table of the LCA. So the LCA will send multiple unicast PDP (using source routing) to the other LCAs in its zone that have adjacent LCAs in neighboring zones. For example, LCA_{24} will unicast the following PDP for LCA_{21}:

$$LCA_{24} \rightarrow LCA_{21} : [\text{PDP},\ IP_S,\ N_S,\ (IP_H,\ IP_{LCA21}),\ CertLZ_2,\ IP_D)\ K_{LCA24-},\ Cert_{LCA24}$$

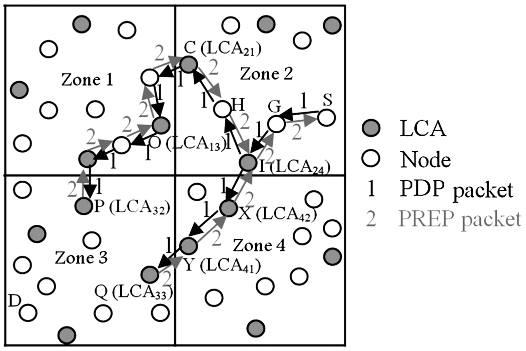


Figure 5. Authenticated location service.

Each LCA in that zone will send this PDP to its adjacent LCA in the neighboring zone. This PDP will be sent using unicast if the adjacent LCA can be reached in one hop. LCA_{24} for example will send:

$$LCA_{24} \rightarrow LCA_{42} : [\text{PDP},\ 2,\ N_{Z2}, IP_{LCA42}, CertLZ_2,\ IP_D]\ K_{LCA24-},\ Cert_{LCA24}$$

If the adjacent LCA is not within the transmission range of the first LCA, PDP will be sent using restricted directional flooding. LCA_{21} for example will send:

$$LCA_{21} \rightarrow \text{broadcast} : [\text{PDP}, 2, N_{Z2}, IP_{LCA13}, P_{LCA13},\ CertLZ_2, IP_D)\ K_{LCA21-},\ Cert_{LCA21}$$

LCA_{13} position (P_{LCA13}) is included in the request because nodes in zone 2 are not aware of the position of LCAs in zone 1. Now, each LCA in the neighboring zones will check if it has received the packet from other LCAs in its zone, then it will drop the packet. Else, it unicasts PDP to the other LCAs in its zone; those have adjacent LCAs in neighboring zones. These steps will be repeated until the PDP packets reach LCAs (LCA_{32} and LCA_{33} in Figure 5) having the destination node in their authentication tables. These LCAs, in turn, will unicast a PREP back along the reverse path to source. Suppose that LCA_{41} can be reached within one hop from LCA_{33}, then LCA_{92} will unicast the following PREP:

$$LCA_{33} \rightarrow LCA_{41} : [\text{PREP},\ 2,\ N_{Z2}, IP_{LCA41},\ P_D, CertLZ_3]\ K_{LCA33-},\ Cert_{LCA33}$$

This packet will be unicast through reverse path till reaching the source node. All the position discovery steps are carried out using the authentication steps used with ARAN protocol. Each node along the PDP path and the reverse (PREP) path validates the previous node's signature, removes the previous node's certificate and signature, signs the original contents of the packet and appends its own certificate. There is only one difference between the behavior of the nodes upon receiving a request or a replay. When a node receives a PDP, it records the previous node's IP address and forwards sending the packet; however, upon receiving a PREP packet, it forwards the replay back to the predecessor from which it received the original request.

3.5. Route discovery and setup

This section explains the needed steps to perform route discovery and setup. After receiving the destination's position (whether local or external one), the source starts instantiating a route to the destination by sending a RDP. This is carried out using restricted directional flooding:

$$\text{S} \rightarrow \text{broadcast} : [\text{RDP}, \ \text{N}_\text{S}, \ \text{IP}_\text{D}, \ \text{P}_\text{D}, \ \text{D}_\text{D}] \ \text{K}_{\text{S}-}, \ \text{Cert}_\text{S}$$

N_S is used to uniquely identify a RDP coming from a specific source. When the destination receives the first RDP, it unicasts a Route REPly (RREP) packet back along the reverse path to the source. Let the first node that receives the RREP sent by *D* be *C*:

$$D \rightarrow C : [\text{RREP}, \ \text{IP}_\text{S}, \ \text{N}_\text{S}] \ \text{K}_{D-}, \ \text{Cert}_D$$

All the route discovery steps are carried out using the authentication steps used with ARAN protocol. Each node along the RDP path and the reverse (REP) path validates the previous node's signature, removes the previous node's certificate and signature, signs the original contents of the packet and appends its own certificate. There is only one difference between the behavior of the nodes upon receiving a request or a replay. When a node receives a RDP, it records the previous node's IP address and forwards sending the packet; however, upon receiving a RREP, it forwards the replay back to the predecessor from which it received the original request.

3.6. Route maintenance

ARANz is an on-demand routing protocol; accordingly, nodes keep track of whether routes are active or not. When no data are received on an existing route, the route is simply deactivated. Data received on an inactive route cause nodes to generate an ERRor (ERR) packet. Nodes also use ERR packets to report links in active routes that are broken owing to node movement. All ERR packets must be signed.

3.7. Data transmission

After finishing route instantiation, source commences sending 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. Consequently, as in ARAN, data packets exchanged between nodes are not signed and do not have attached certificates. Hence, each node simply relays a data packet as is to its successor in the route obtained during the route initiation process. However, to ensure the data privacy and prevent other trusted nodes from reading the data itself, the data may be encrypted using the public key of the destination that the source may obtain through position discovery phase.

4. PERFORMANCE EVALUATION

In this section, the performance of ARANz is studied and compared with existing protocols. We shall compare our protocol with the original ARAN protocol because our protocol is based on it. Besides, we will also consider AODV protocol for comparison issues because AODV is often considered as a benchmark for evaluating the performance of ad hoc routing protocols and because ARAN has been proposed based on it. We begin this section with a summary of the properties of the discussed protocols. After that, an analysis of their robustness in the presence of different attacks is provided in subsection 4.2, whereas subsection 4.3 provides a preliminary simulated performance evaluation.

4.1. Summary of the evaluated protocols

Table VII summarizes properties of the discussed protocols. Both AODV and ARAN are reactive topology-based routing protocols that use broadcasting in the route discovery process, whereas ARANz is a restricted directional flooding position-based routing protocol. AODV does not define any security requirements and inherently trusts all participants. On the contrary, ARAN and ARANz use cryptographic certificates to prevent most of the attacks against ad hoc routing protocols and detect erratic behavior. ARANz also aims to achieve high level of security and avoid single point of attack by distributing trust among multiple LCAs. All three protocols are loop-free and can be implemented at any network density.

Both AODV and ARAN are reactive topology-based routing protocols that use broadcasting in the route discovery process, whereas ARANz is a restricted directional flooding position-based routing protocol. AODV does not define any security requirements and trusts all participants.

On the contrary, ARAN and ARANz use cryptographic certificates to prevent most of the attacks against ad hoc routing protocols and detect erratic behavior. Moreover, ARANz aims to achieve high level of security and avoid single point of attack problem by distributing trust among multiple LCAs. All 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 at any network density.

Ad Hoc On-demand Distance Vector selects the path with the minimum number of hops. ARAN and ARANz do not guarantee the shortest path, but they offer the quickest path that is chosen by the RDP that reaches the destination first. Simulations result in [1] showed that the average path length for AODV and ARAN are almost identical. This indicates that even though ARAN does not explicitly seek the shortest paths, the first RDP 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 have the same criterion.

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 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

Table VII. Characteristics of the presented protocols.

Performance parameter	AODV	ARAN	ARANz
Type	Topology-based (reactive)	Topology-based (reactive)	Position-based (restricted directional flooding)
Secure	No	Yes	Yes
Route discovery sending mechanism	Route discovery packets are flooded to all nodes in the network	Route discovery packets are flooded to all nodes in the network	Intermediate nodes broadcast route discovery packet only if they are closer to the destination than the previous hop
Main idea/contribution	Initiating a route discovery process only when the route is needed	Protecting routing packets against attacks from malicious nodes in managed-open environments	Solving scalability as well as single point of compromise and failure problems existing in ARAN protocol
Proposal	Uses next hop information stored in the nodes of the route with the least number-of-hop field	• Provides authentication of route discovery, setup and maintenance • Uses cryptographic certificates to prevent most security attacks that face ad hoc routing protocols • Routing messages are authenticated at each hop from source to destination, as well as on the reverse path from destination to source	• Divides area into zones and introduces multiple LCAs in each zone • Requires sending a PDP if the position of the destination is unknown • Uses restricted directional flooding to forward RDP • Provides authentication of position update and discovery as well as route discovery, setup and maintenance • Uses cryptographic certificates to prevent most security attacks that face ad hoc routing protocols
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 (multiple LCAs in each zone)
Synchronization	No	No	Yes
Robustness	High	Low	Medium

compromised one. Using multiple LCAs in ARANz, on the other hand, raises the need to keep them synchronized.

Ad Hoc On-demand Distance Vector and ARAN are more robust in the route discovery phase than ARANz because 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, roughly, have the same robustness because the failure of an individual node might result in packet loss and setting up a new route. ARANz tries to achieve higher robustness compared with ARAN by distributing trust among different LCAs; multiple LCAs should collaborate to issue certificates for the nodes inside a particular zone. Hence, a failure of a single LCA (or even multiple LCAs) will not affect updating nodes' certificates because other LCAs in its zone are able to discover its failure and elect another LCA to replace it. However, in ARAN, the CA is a vital of the network, and its failure prevents 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 low and medium, respectively.

4.2. ARAN and ARANz security analysis

Just like ARAN protocol, ARANz uses cryptographic certificates to prevent most of the security attacks that ad hoc routing protocols face. It introduces authentication, message integrity and non-repudiation as part of a minimal security policy for the ad hoc environment. Moreover, confidentiality in ARANz is ensured if important data are encrypted with the destination's public key.

Because all ARAN packets must be signed, a node cannot participate in routing without authorization from the CA. This access control therefore relies on the security of the CA, the authorization mechanisms employed by the CA, the strength of issued certificates and the revocation mechanism. Hence, this CA is a single point of attack, and it is a big concern to keep this CA uncompromised. In ARANz, a node is allowed to participate in routing after gaining authorization from the LCAs of its zone. Even if one LCA is compromised, the revocation mechanism discussed in subsection 3.3.5 can be executed to exclude this LCA from the network and elect a new one. One may think that introducing multiple LCAs may cause compromising the network if any of them is compromised. However, as mentioned in subsection 3.3.5, security of the whole network is compromised only if three LCAs of a particular zone are compromised at the same time without being able to identify them as compromised. In this case, these LCAs can collaborate together to issue certificates for untrusted nodes in their zone. Consequently, a higher level of availability is achieved by ARANz owing to avoiding single point of attack problem. On the contrary, the centralized CA in ARAN protocol results in lower availability because the compromise of this CA affects the security of the entire network.

The following is an analysis of the robustness of ARAN and ARANz in the presence of different attacks introduced in subsection 2.3:

- Passive attacks: detecting passive attacks is very difficult because the operation of the network itself is not affected. One way of overcoming such problems is to use powerful encryption mechanisms to encrypt the data being transmitted, thereby making it impossible for eavesdroppers to obtain any useful information from the data overheard [20]. Both ARAN and ARANz uses cryptographic operations to protect control packets from eavesdropping.
- Active attacks: both protocols are robust against most active attacks, as shown in the following discussion:
- Spoofed route signaling: all request packets, in ARAN as well as ARANz, are signed with the source's private key and contain its certificate. Similarly, reply packets include the destination's certificate and signature, ensuring that only the destination can respond to a particular request. This prevents impersonation attacks where either the source or destination is spoofed.
- Fabricated routing messages: ARAN and ARANz do not prevent fabrication of routing messages, but they offer a deterrent by ensuring non-repudiation because all routing messages must contain the sender's certificate and signature. Therefore, a node that injects false messages into the network may be excluded from future route discovery processes.
- Alteration of routing messages: both protocols specify that all fields of request and reply packets remain unchanged between source and destination. Because both packet types are signed by the initiating node, any alterations in transit would be detected, and the altered packet would be subsequently discarded. Thus, modification attacks are prevented in both protocols.

Tables VIII and IX give summary of the security requirements satisfied by both protocols as well as different attacks they defend against.

4.3. Simulated network performance

GloMoSim is used as a simulation tool to evaluate the performance of AODV, ARAN and ARANz protocols. AODV is already implemented in GloMoSim, so two new models called "ARAN" and "ARANz" have been added to GloMoSim to simulate ARAN protocol and our new protocol, respectively.

Node transmission range of 250 m is used. The initial nodes positions are chosen randomly with node density of 60 nodes/km^2. After that, all nodes are allowed to move according to the random waypoint mobility model. In this model, each node travels to a randomly selected location at a configured speed and then pauses for a configured pause time, before choosing another random location and repeating the same steps.

802.11 MAC layer and Constant Bit Rate (CBR) traffic over User Datagram Protocol have been used. Source and destination pairs are chosen randomly in both local and external communications. Five CBR sessions are simulated

Table VIII. ARAN and ARANz security analysis.

Criterion	ARAN	ARANz
Secure extension of	AODV	AODV
Basic security mechanism	Certificates and time stamps	Certificates and time stamps
Central trust	Yes (CA)	No (multiple LCAs in each zone)
Availability	Low	Medium
Authentication	Yes	Yes
Confidentiality	No	Yes, if data are encrypted with destination's public key
Integrity	Yes	Yes
Non-repudiation	Yes	Yes
Anonymity	No	No

Table IX. ARAN and ARANz robustness against existing attacks.

Type	Attack	Robust against
Passive attacks	Eavesdropping	Yes
Active attacks	Impersonation	Yes
	Fabrication	No, but provides non-repudiation
	Modification	Yes

in each run. Each session generates 1000 data packets of 512 bytes each at the rate of four packets per second. Local communication percentage of 60% has been used. Hence, in each run, three of the five CBR sessions are local and the other two are external. The motive behind choosing this percentage is that the chance for a node to communicate with a node that is close to it is higher than communicating to a node that is far away from it.

The key distribution in ad hoc networks is beyond the scope of our work, and it deserves further devoted research. Accordingly, for simulating ARAN and ARANz, we assumed that the key distribution procedure is finished, so that all hosts can examine the genuineness of the signed packets. ARAN and ARANz are simulated using a 512-bit key and 16-byte signature. These values are reasonable to prevent compromise during the short time that nodes spend away from the certificate authority and in the ad hoc network [1].

For either protocol, a routing packet processing delay of 1 ms is assumed. This value was obtained through field testing of the AODV protocol implementation [3]. Additionally, a processing delay of 2.2 ms is added to account for the cryptographic operations for ARAN and ARANz. This value is adopted from [1], which they obtained through the implementation testing of measuring processing routing messages of ARAN for both a laptop and a handheld computer. Also, a random delay between 0 and 10 ms is introduced before the retransmission of a broadcast packet to minimize collisions.

To have a consistent comparison of results, a basic version of AODV is used, which does not include optimizations such as the expanding ring search and local repair of routes.

The effect of three important parameters of ad hoc network has been tested. These parameters are node mobility speed, area size and malicious node percentage. For each parameter, five performance metrics are evaluated. These metrics are as follows:

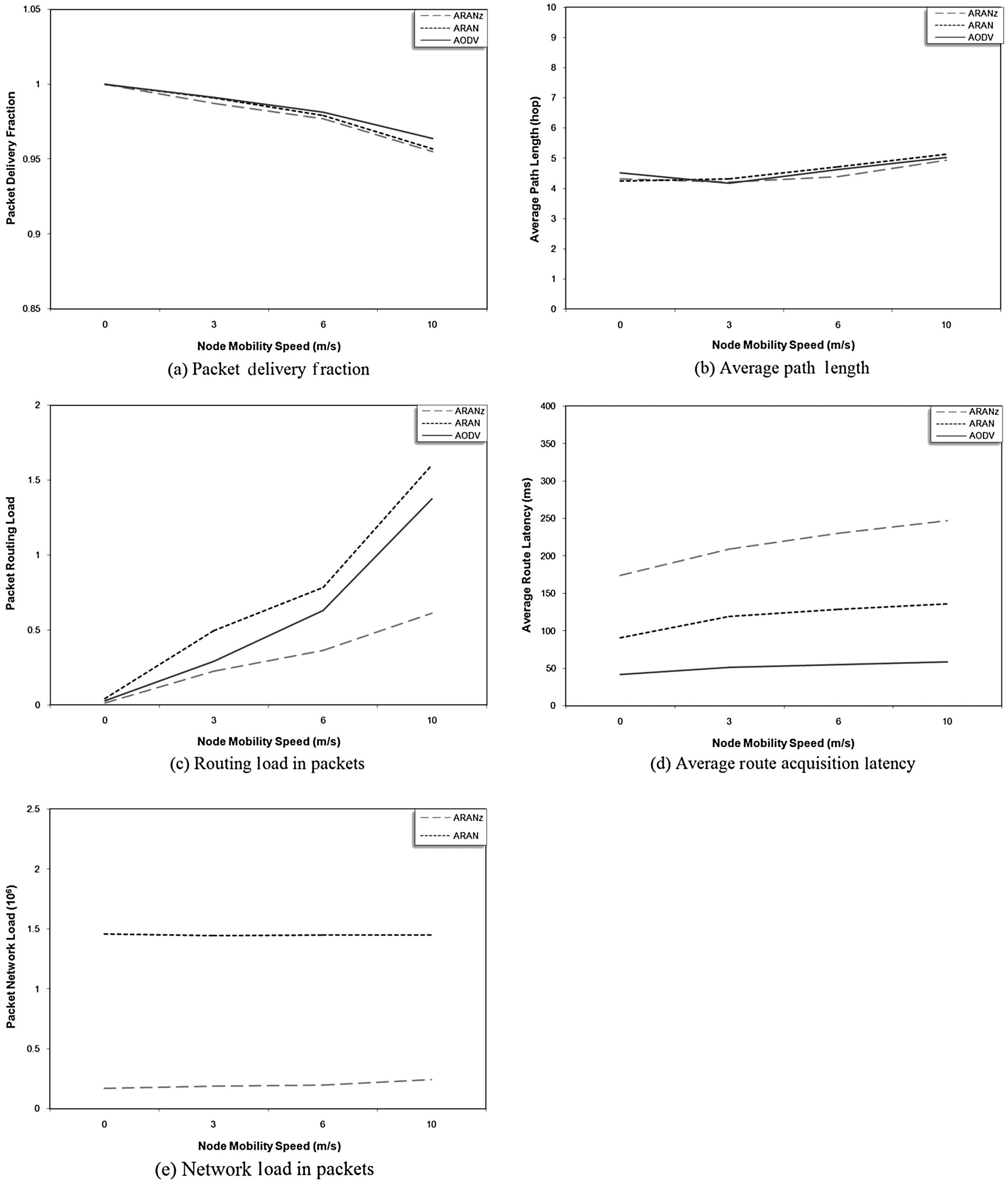
1. Packet delivery fraction: the fraction of data packets generated by the CBR sources that are received by their destinations. This evaluates the ability of the protocol to discover and maintain routes.
2. Average path length: the average length of the paths discovered by the protocol. It is calculated by averaging the number of hops taken by each data packet to reach the destination.
3. Packet routing load: the ratio of routing packets to delivered data packets. Routing packets are those sent during the location service phase as well as route instantiation and maintenance phase. The transmission at each hop along the route also is counted in the calculation of this metric.
4. Average route acquisition latency: the average delay needed for discovering a route to a destination. It is defined in ARAN and AODV as the average delay between sending a route request/discovery packet by a source and receiving the first corresponding route reply packet. In ARANz, it is defined as the average delay for both discovering position of the destination as well as initiating a route to it. If a request timed out and needed to be retransmitted, then the sending time of the first transmission is used for calculating the latency.
5. Packet network load: the overhead packets resulted from constructing and maintaining network structure as well as updating nodes' positions and certificates. It is calculated in ARANz as the summation of all packets sent during the setup and maintenance phases. On the contrary, it is calculated in ARAN as the summation of all packets sent to update nodes certificates. The transmission at each hop along the paths also is counted in the calculation of this metric. Related to AODV, it is a flat non-secure topology-based routing protocol; that is, there are no packets sent neither to maintain network structure nor to update nodes' positions or certificates. As such, packet network load of AODV is excluded from the figures.

For the following figures, each point is an average of five simulation runs with identical configuration but different randomly generated numbers.

4.3.1. Node mobility speed effect

To study the effect of node mobility speed a 2 × 2-km network is considered. This terrain is used because it is considered as a moderate-sized ad hoc network. This network contains 240 well-behaved nodes and is divided into four zones. Simulations are run with 0-m/s, 3-m/s, 6-m/s and 10-m/s speeds with pause time fixed at 30 s.

As shown in Figure 6(a), the packet delivery fraction obtained using ARANz protocol is higher than 95% in all scenarios. This suggests that ARANz is highly effective in discovering and maintaining routes for delivery of data packets even with relatively high node mobility. As we can also see from the figure, packet delivery fraction for ARANz and ARAN is identical to that for AODV in the low node mobility, but it is slightly less when mobility increases. This is because of higher packet processing and authentication



(a) Packet delivery fraction

(b) Average path length

(c) Routing load in packets

(d) Average route acquisition latency

(e) Network load in packets

Figure 6. Node mobility speed effect.

delay at each node in the case of using ARAN and ARANz protocols. In other words, longer time means higher probability for losing the link connection because of node movement, which results in dropping some packets. This time will be longer if node mobility resulted in increasing the distance between source and destination, which accordingly increases number of hops.

Even though ARAN and ARANz do not explicitly seek the shortest paths, the first RDP to reach the destination usually travels along the shortest path (as shown in Figure 6(b)). Hence, it is obvious that ARAN and ARANz are as efficient as AODV in discovering the shortest paths. The average path length for the three protocols increases slightly with increasing node mobility. This mobility may result in separating source and destination nodes from each other and producing longer paths.

As shown in Figure 6(c), ARANz has the minimum packet routing load. ARANz does not broadcast the RDP to whole area. In ARANz, the RDP is sent using restricted directional flooding towards the destination; this is the reason behind reducing the overall routing load. Even PDPs are sent using restricted directional flooding or source routing. Also, it is clear from the figure that ARAN has higher packet routing load than AODV, although both protocols broadcast route request/discovery to the entire network. This is because of two reasons. The first is because ARAN has higher packet processing and authentication delay at each node, which in turn, increases the chance of a link break because of node movement. This break causes the source to reinitiate a new RDP, which increases the overall routing load for ARAN. Second, if an intermediate node in AODV has a valid route towards destination, then it can answer with a RREP to the source. Hence, there is no need to rebroadcast the RREQ to its neighbors, which in turn reduces the overall AODV routing load.

To study the effect of node mobility on packet routing load, let us refer again to Figure 6(c). It is clear from the figure that routing load for the three protocols increase with increasing node mobility. This is because increasing mobility will increase the chance for losing the link connection and reinitiating RDP, which increases overall packet routing load for the three protocols.

Figure 6(d) shows that the average route acquisition latency for ARAN and ARANz protocols is higher than that for AODV. While processing ARAN and ARANz routing control packets, each node has to verify the digital signature of the previous node and then replace this with its own digital signature, in addition to the normal packet processing as carried out by AODV. This signature generation and verification causes additional delay at each hop, so average route acquisition latency increases. Moreover, although some other position-based routing protocols assume that the position of the destination is known previously to the source or that nodes periodically broadcast their positions to all other nodes in the network, ARANz has high latency because it takes into consideration the time required to make an enquiry about the position of the destination.

Figure 6(d) also shows that average route acquisition latency for the three protocols slightly increase with increasing node mobility. This mobility may make the nodes far away from each other that produce longer paths, thus more latency.

As shown in Figure 6(e), packet network load for ARANz is much less than that for ARAN. The main reason behind this difference is that nodes in ARAN are unaware of CA position; hence, all periodic certificate update request packets sent from nodes to CA are broadcasted to the entire network. In ARANz, however, certificate update request packets as well as position update packets are sent from nodes towards the nearest LCA to themselves using restricted directional flooding. After that, these packets are forwarded from the nearest LCA to other LCAs in its zone using source routing. Even packets sent upon updating LCA position or electing a new LCA, they are sent only to nodes in the intended zone and adjacent LCA in the immediate neighboring zone. Moreover, packets sent in case of node movement to a neighboring zone are sent using source routing or restricted directional flooding. Additionally, node failure packets and LCA synchronization packets are sent only to LCAs in the network using LCA flooding. Finally, the only two packets that are broadcasted to the entire network are NETSET packet during the network setup and CNODE packet to indicate misbehaving of a particular node.

Packet network load for ARANz increases slightly as the node mobility increases. Frequent node mobility results in increasing number of packets sent for updating node positions as well as electing new LCAs. In ARAN, however, certificate update request packets are broadcasted to the entire network regardless of node mobility speed.

4.3.2. Area size effect

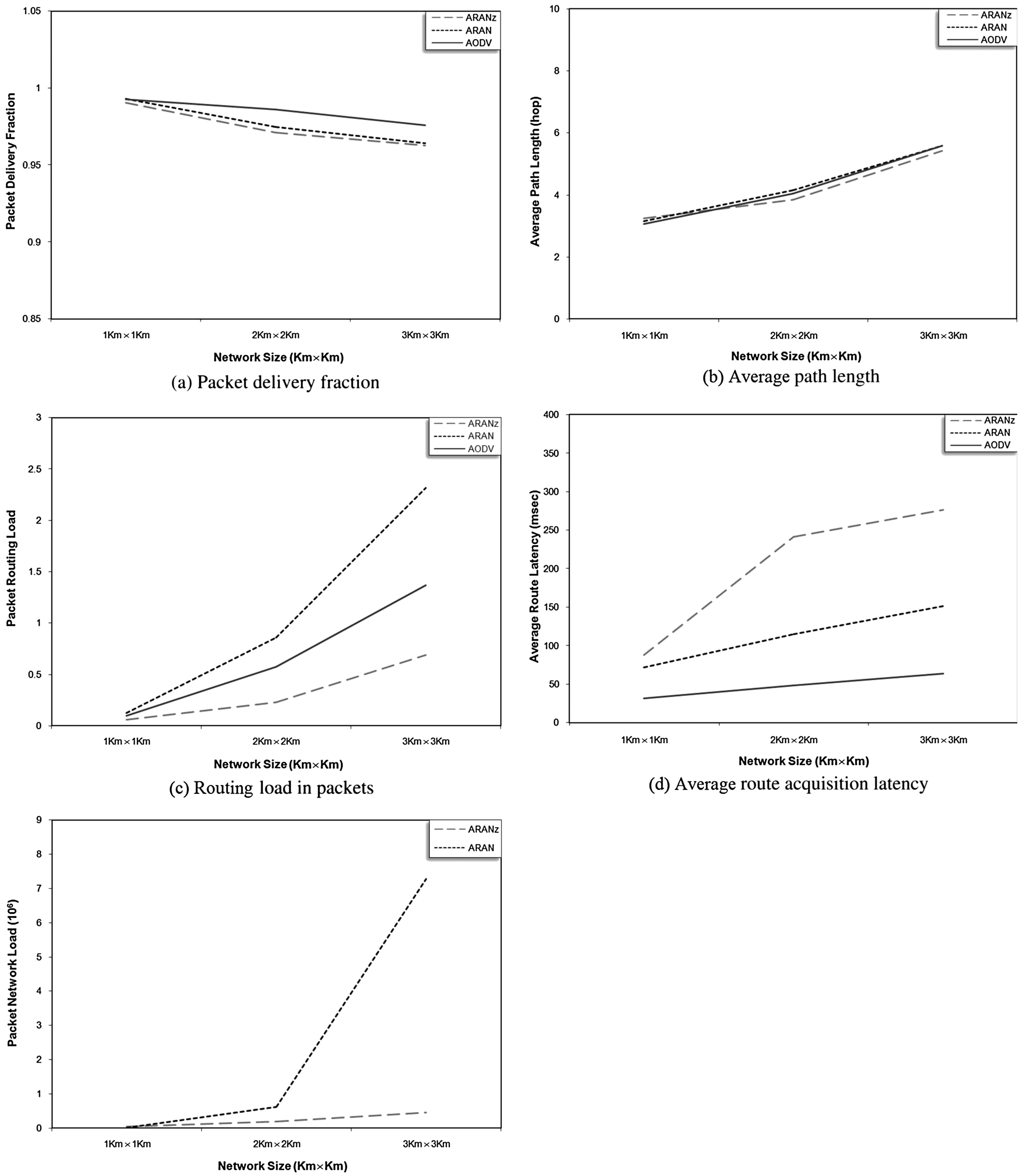
To study the effect of area size, three networks of 1 × 1-km, 2 × 2-km and 3 × 3-km area sizes are tested. These networks are divided into multiple zones each of 1 × 1 km. Simulations are run with 60 nodes/km^2. These nodes are moving with a speed of 5 m/s, as it is considered as a moderate speed, with pause time fixed at 30 s. All nodes are considered to be well behaved.

As shown in Figure 7(a), packet delivery fraction for ARAN and ARANz is slightly less than that for AODV especially in the large area network. This is because of higher packet processing and authentication delay at each node in the case of using ARAN and ARANz protocols. This delay increases the probability for losing the link connection because of node movement, which results in dropping some packets. Also, the fraction of data packets delivered decreases with increasing the area size for the three protocols because of higher number of nodes that the packet passes through, which increases the probability of link break.

Figure 7(b) shows that the average path length for the three protocols is almost identical for a specified network size. This indicates that ARAN and ARANz are as efficient as AODV in discovering the shortest paths. Also, it is clear that the average length of the discovered paths increases with increasing the area size because of higher number of nodes that the packet passes through if the source and destination are apart from each other, which means longer paths.

As shown in Figure 7(c), packet routing load for the three protocols increase with increasing area size because of higher probability of link break that requires reinitiating a RDP. ARANz still has the minimum packet routing load as a result of using restricted directional flooding in forwarding RDP. Additionally, it is clear that ARAN has higher routing load than AODV because ARAN has higher packet processing and authentication delay, which increases the chance of a link break and reinitiating a new RDP.

Figure 7(d) shows that the average route acquisition latency for the three protocols increases with increasing area size because of increasing number of nodes that the control packets pass through. As the figure shows, ARANz has the highest latency because of time required for



(a) Packet delivery fraction

(b) Average path length

(c) Routing load in packets

(d) Average route acquisition latency

(e) Network load in packets

Figure 7. Area size effect.

position enquiry process. Moreover, average route acquisition latency for ARAN is higher than that for AODV because of higher ARAN packet processing and authentication delay.

As shown in Figure 7(e), packet network load for ARAN and ARANz are almost identical when a network of 1 × 1-km size is used. ARANz deals with this network size as one zone; hence, some packets such as those sent upon updating LCA position or electing a new LCA are broadcasted to the entire network. In large area networks, however, network load for ARAN becomes much higher than that for ARANz, reaching to more than 10 times in 3 × 3-km area network. This large gap results from broadcasting certificate update request packets to the entire network upon using ARAN. The packet network load for both protocols increases as the network size increases. Larger network size results in increasing number of packets sent for updating nodes' positions and certificates.

4.3.3. Malicious node percentage effect

The experiments described in the previous sections compare the performance of the three protocols where all nodes are considered well behaved. This section tries to investigate the efficiency of our secure routing protocol in detecting malicious nodes. The effect of malicious node behavior is studied on 2 × 2-km network that contains 240 nodes and divided into four zones. These nodes are moving with a speed of 5 m/s. Five CBR sessions are simulated in each run: three of them are local and two are external. Simulations are run with 0%, 10%, 20% and 40% malicious node percentages. These malicious nodes are selected randomly.

The malicious behavior simulated in this scenario is an example of the modification attack. Whenever a malicious node receives a route request/query or route reply, it draws a random number between 0 and 1. If the drawn number is less than 0.5, then it illegally resets the hop count field to 0, pretending to be only one hop away from the source or destination. Otherwise, the control packet is forwarded without modification.

The obtained results show that packet delivery fraction, packet routing load and packet network load for the three protocols are, roughly speaking, not affected by malicious node percentage. Hence, these three metrics have been excluded from the presented results and another three metrics have been added for this experiment, which are as follows:

1. Malicious route percentage: the fraction of the used routes that have malicious nodes within them. It is calculated as number of routes passing through malicious nodes over the total number of routes.
2. Compromised node percentage: the percentage of nodes that have been considered as compromised as a result of recognizing their misbehavior.
3. Packet malicious load: the overhead packets resulted from sending misbehavior detection packets such as MNODE and CNODE packets. The transmission at each hop along the paths is also counted in the calculation of this metric.

It is clear that the last two metrics are specified for ARANz protocol because neither ARAN nor AODV has a misbehavior detection system.

It is clear from Figure 8(a)–(b) that the average path length and route acquisition latency for ARAN and ARANz are roughly not affected by the simulated malicious node percentage, whereas these two metrics for AODV increase as malicious node percentage increases. This is because malicious nodes can exploit AODV so that the non-shortest paths are selected, whereas such exploitation is not possible with ARAN and ARANz.

Figure 8(c) shows that the malicious route percentage is increased for the three protocols when malicious node percentage is increased. However, upon using AODV, a much larger fraction of routes have malicious nodes within them. When the malicious node resets the hop count field to 0, it forces the selected routes to pass through itself because AODV selects the shorter path. ARAN and ARANz, on the contrary, cannot be exploited in this fashion. The selected route could pass through a malicious node but not forced to this. Results also show that ARANz has the minimum malicious route percentage because of detecting and excluding malicious nodes from the future used routes. Excluding malicious nodes from the used routes reduces the probability of performing attacks during the forwarding process such as replaying, modifying and dropping data packets.

Referring to Figure 8(d)–(e), it is apparent that compromised node percentage and packet malicious load for ARANz increase as the malicious node percentage increases. This suggests that ARANz is efficient in identifying and isolating modification attackers.

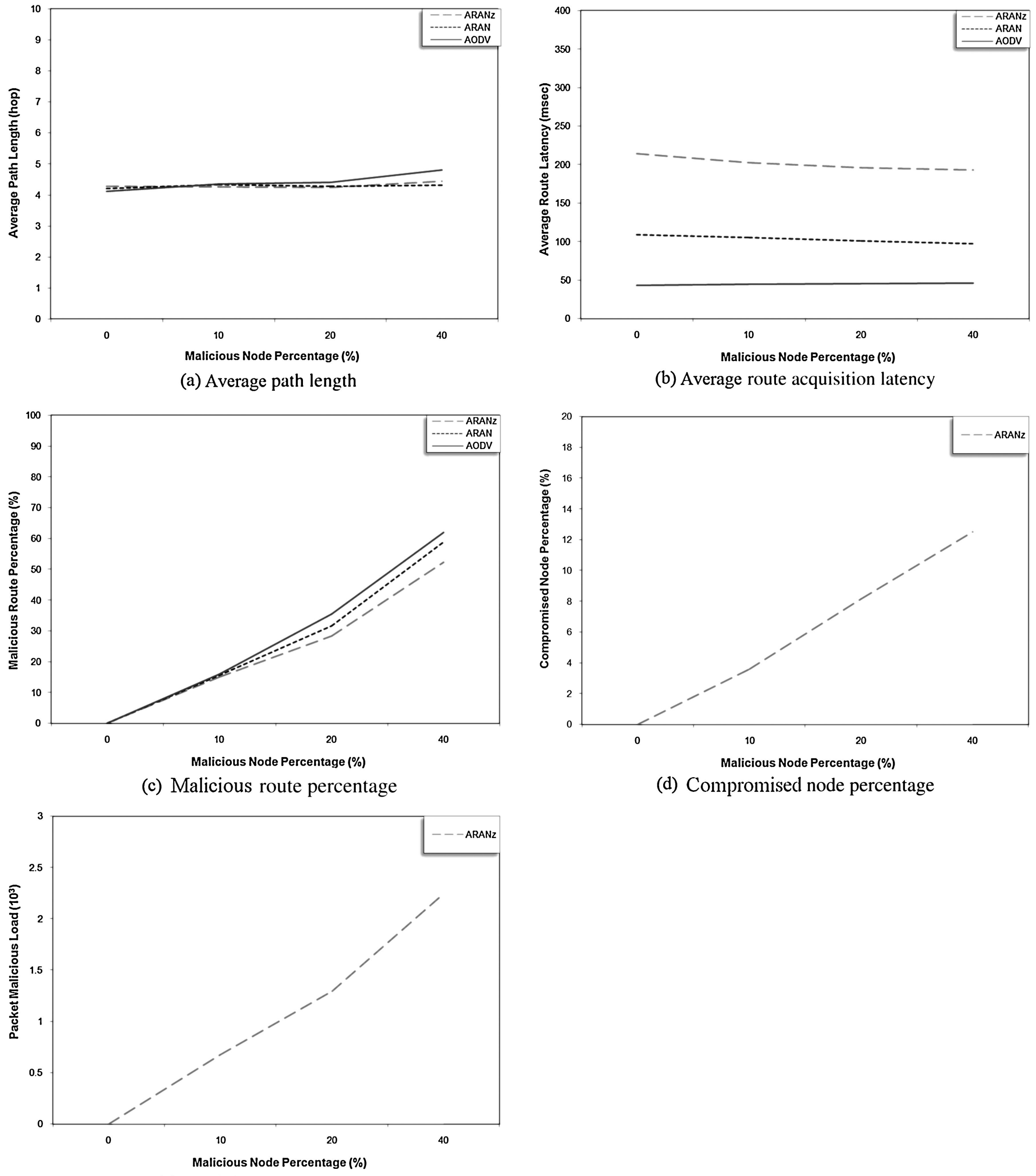
5. DISCUSSION

Ad Hoc On-demand Distance Vector is a non-secure reactive routing protocol; hence, it has less processing overhead compared with ARAN and ARANz because nodes in AODV do not apply cryptographic operations such as validating the previous node's signature, signing the routing packets and appending certificates. AODV uses broadcasting in the route discovery phase that increases its robustness against nodes' failure on one hand, whereas it increases packet overhead on the other hand. This is because the route request packet is sent to all nodes in the network. Because of this, AODV is not a scalable protocol.

Like AODV, ARAN is a reactive routing protocol that uses broadcasting in the route discovery process. ARAN uses cryptographic certificates to prevent most of the attacks against ad hoc routing protocols such as modification, impersonation and fabrication as well as to detect erratic behaviors such as the use of invalid certificates, improperly signed packets and misuse of some packets. However, the usage of these certificates increases the route acquisition latency as well as packet and processing overheads compared with AODV. These increased latency and overhead are because of the encryption/decryption processes along with route request broadcast. ARAN also suffers from the centralized trust and load, that is, single point of attack and failure. Similar to AODV, ARAN has scalability problem owing to using one certificate authority server that can be the operation bottleneck.

With ARANz, a scalable and secure solution can be achieved. Adopting the authentication methods used in ARAN, ARANz is a secure routing protocol. Additionally, via dealing with the network as zones and using restricted directional flooding, our new model aims to exhibit better scalability and performance. As opposite to ARAN, ARANz distributes load and trust by dividing the area into zones and introducing multiple certification authorities (i.e., LCAs) in each zone. Distributing load and trust helps in achieving the following:

(a) High level of security by avoiding single point of attack problem. In ARANz, security of the network



(a) Average path length

(b) Average route acquisition latency

(c) Malicious route percentage

(d) Compromised node percentage

(e) Malicious load in packets

Figure 8. Malicious node percentage effect.

is compromised only if three LCAs in the same zone are compromised at the same time.

(b) High level of robustness owing to avoiding single point of failure problem. A failure of a single LCA in ARANz does not affect updating nodes' certificates because other LCAs in its zone are able to discover its failure (via periodic certificate update process) and elect another LCA to replace it. However, in ARAN, the CA is vital to the network and its failure prevents all nodes from updating their certificates.

Using multiple LCAs in ARANz, on the contrary, comes up with a need to keep them synchronized.

Our simulation results show that ARAN and ARANz are as effective as AODV in discovering and maintaining not only routes but also the shortest paths. It is obvious from the simulation results that ARANz has achieved the scalability issue by maintaining the minimum packet routing load even with large networks and high node mobility. ARANz reduced packet routing load is a natural result of sending RDP using restricted directional flooding towards the destination.

The cost of ARAN and ARANz security is higher latency in route discovery because of cryptographic computation that must occur. ARANz reduced packet routing load comes in the cost of higher latency in route discovery because of the time required to obtain destination's position.

6. CONCLUSIONS

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 hierarchal and distributed routing algorithm, which improves performance and scalability of the routing protocol by dividing the area into zones. ARANz aims to achieve robustness, increase network security and solve the single point of failure and attack problems by introducing multiple LCAs. Our ARANz also aspires to exhibit better scalability, performance and robustness against frequent topological changes via the restricted directional flooding position-based routing protocols.

Our preliminary simulations show that ARANz is highly effective in discovering the shortest paths and maintaining secure routes even with relatively high node mobility, large network size and large percentage of malicious nodes. Moreover, ARANz has achieved the scalability issue by maintaining the minimum packet routing load in all presented scenarios compared with AODV and ARAN protocols. The cost of ARANz is higher latency in route discovery on account of the time required for authentication and packet processing as well as obtaining destination's position.

7. FUTURE WORKS

Our next tasks are to evaluate the effectiveness of the protocol in dealing with security issues considering different number of malicious nodes existing in the network and performing different types of attacks. We also aim to test ARANz scalability considering different node densities, different zone sizes, different number of LCAs in each zone and different local communication percentages. Comparisons will then be performed with other existing secure routing protocols especially secure AODV extensions such as [10–12]. Last but not least, in our current work, we have considered that nodes are evenly geographically distributed. It is one of our future tasks to consider the case when some regions of the network have very few nodes and some others have much more.

ACKNOWLEDGEMENTS

We want to thank the editors and the anonymous reviewers for their valuable comments on previous drafts of this manuscript. This work was carried out under the VotF University Malaya fund no. FS132/2008C and PPP University Malaya fund no. PS091/2009A and PS410/2010B.

REFERENCES

1. Sanzgiri K, LaFlamme D, Dahill B, Levine B, Shields C, Belding-Royer E. Authenticated routing for ad hoc networks. *IEEE Journal On Selected Areas In Communications* 2005; **23**(3):598–610.
2. Qabajeh L, Mat Kiah ML, Qabajeh M. A scalable and secure position-based routing protocol for ad-hoc networks. *Malaysian Journal of Computer Science* 2009; **22**(2):99–120.
3. Perkins C, Royer E. Ad hoc on-demand distance vector routing. *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*. New Orleans, USA, 1999; 90–100.
4. Beijar N. Zone Routing Protocol (ZRP). Networking Laboratory, Helsinki University of Technology, Finland. Available from: http://www.netlab.hut.fi/opetus/s38030/k02/Papers/08-Nicklas.pdf, 1998 [accessed on 31 March 2012]
5. Lin T. Mobile Ad-hoc Network Routing Protocols: Methodologies and Applications. PhD Thesis, Faculty of the Virginia Polytechnic Institute and State University, Blacksburg, Virginia. Available from: http://scholar.lib.vt.edu/theses/available/etd-03262004-144048/unrestricted/Tao_PhD_Dissertation.pdf, 2004.
6. Abolhasan M, Wysocki T, Dutkiewicz E. A review of routing protocols for mobile ad hoc networks. *Elsevier Ad Hoc Networks* 2004; **2**(1):1–22.
7. Cao Y, Xie S. A position based beaconless routing algorithm for mobile ad hoc networks. *Proceedings of the IEEE International Conference on Communications Circuits and Systems*. Hong Kong, China, 2005; 303–307.
8. Li H, Singhal M. A secure routing protocol for wireless ad hoc networks. *Proceedings of the 39th IEEE Annual Hawaii International Conference on System Sciences*. Hawaii, USA, 2006; 225a-225a.
9. Mizanur Rahman Sk, Mambo M, Inomata A, Okamoto E. An anonymous on-demand position-based routing in mobile ad hoc networks. *Proceedings of the International Symposium on Applications and the Internet*. Arizona, USA, 2006; 300–306.
10. Zapata M. Secure ad hoc on-demand distance vector routing. *ACM Mobile Computing and Communications Review* 2002; **6**(3):106–107.

11. Cerri D, Ghioni A. Securing AODV: the A-SAODV secure routing prototype. *IEEE Communications Magazine* 2008; **46**(2):120–125.
12. Li Q, Zhao M, Walker J, Hu Y, Perrig A, Trappe W. SEAR: a secure efficient ad hoc on demand routing protocol for wireless networks. *Security and Communication Networks* 2009; **2**(4):325–340.
13. Karp B, Kung H. GPSR: greedy perimeter stateless routing for wireless networks. *Proceedings of the 6th ACM/IEEE Annual International Conference on Mobile Computing and Networking*. Massachusetts, USA, 2000; 243–254.
14. Wu X. VPDS: virtual home region based distributed position service in mobile ad hoc networks. *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*. Ohio, USA, 2005; 113–122.
15. Ko Y, Vaidya N. Location-aided routing (LAR) in mobile ad hoc networks. *ACM Wireless Network* 2000; **6**(4):307–321.
16. Blazevic L, Buttyan L, Capkum S, Giordano S, Hubaux J, Le Boudec J. Self-organization in mobile ad-hoc networks: the approach of terminodes. *IEEE Communication Magazine* 2001; **39**(6):166–174.
17. Carter S, Yasinsac A. Secure position aided ad hoc routing. *Proceedings of the IASTED International Conference on Communications and Computer Networks*. Cambridge, UK, 2002; 329–334.
18. Song J, Wong V, Leung V. Secure position-based routing protocol for mobile ad hoc networks. *Elsevier Ad Hoc Networks Journal* 2007; **5**(1):76–86.
19. Mahmoud A. Reputed authenticated routing for ad hoc networks protocol (reputed-ARAN) 2005. Master Thesis, Computer Science Department, The American University in Cairo. Available from: www.mbifoundation.com/media/18949/Abdalla%20Mahmoud%20-%20Thesis%20Defense.pdf
20. Murthy C, Manoj B. *Ad Hoc Wireless Networks: Architectures and Protocols*. Prentice Hall Communications Engineering and Emerging Technologies Series: Upper Saddle River, New Jersey, United States, 2004; 213–214 and 475–490. ISBN:013147023X
21. Zhou L, Haas Z. Securing ad hoc networks. *IEEE Networks Special Issue on Network Security* 1999; **13**(6):24–30.
22. Zou C, Chigan C. An anonymous on-demand source routing in MANETs. *Security and Communication Networks* 2009; **2**(6):476–491.
23. Chen H, Xiao Y, Hong X, Hu F, Xie J. A survey of anonymity in wireless communication systems. *Security and Communication Networks* 2009; **2**(5):427–444.
24. Seys S, Preneel B. ARM: anonymous routing protocol for mobile ad hoc networks. *International Journal of Wireless and Mobile Computing* 2009; **3**(3): 145–155.
25. Park Y, Lee W, Rhee K. Authenticated on-demand ad hoc routing protocol without pre-shared key distribution. *Proceedings of the ECSIS Symposium on Bio-inspired, Learning and Intelligent Systems for Security*. Edinburgh, UK, 2007; 41–46.
26. Pirzada A, McDonald C. Reliable routing in ad hoc networks using direct trust mechanisms. In *Advances in Wireless Ad Hoc and Sensor Networks*, Cheng M, Li D (eds). Springer: Verlag, London, United Kingdom, 2008; 133–159. ISBN:0387685650
27. Rifa-Pous H, Herrera-Joancomarti J. Secure dynamic MANET on-demand (SEDYMO) routing protocol. *Proceedings of the 5th Annual Conference on Communication Networks and Services Research*. Fredericton, Canada, 2007; 372–380.
28. Sivrikaya F, Yener B. Time synchronization in sensor networks: a survey. *IEEE Network* 2004; **18**(4): 45–50.