A Scalable Secure Routing Protocol for Managed-Open Mobile Ad-Hoc Networks

Liana Khamis Qabajeh
Information Technology and Computer Engineering Faculty, Palestine Polytechnic University, Hebron, Palestine
Email: liana_tamimi@ppu.edu

***Abstract*—Mobile Ad-hoc networks are getting progressively more popular. In such networks, nodes are continuously moving, hence, seeking an efficient route from a particular source to projected destination is a vital issue. Moreover, choosing a secure route is a tough area to deal with since adversaries may include themselves within these routes unless a strict secure procedure is implemented along with routing protocols. In managed-open environment, such as that formed by peers at a conference, using already established infrastructure as a starting point for assuring security is likely. Thus, this research proposes a novel paradigm of routing protocol named S-Octopus. Through tackling with area as sectors and utilizing restricted directional flooding, S-Octopus tries to achieve improved scalability. Moreover, S-Octopus seeks to increase robustness and alleviate single point of failure and compromise problem via picking up numerous sector certificate authorities. A qualitative comparison among S-Octopus and some other secure Ad-hoc routing models is presented in this paper.**

***Index Terms*—position-based, secure, scalable, routing protocol, Ad-hoc networks, managed-open, S-Octopus**

I. Introduction

One attractive type of wireless networks is the Ad-hoc network since it is a self-regulating multi-hop type [1]. All nodes in the network participate in forwarding packets and are moving rapidly in most cases [2], [3]. Ad-hoc networks may be set up dynamically when needed, since they do not need pre-established infrastructure. Accordingly, they are implemented in different areas, such as emergency situations, community interacting, and search and rescue operations [3]-[6].

Upon implementing such networks, it is a crucial issue to reduce transmission overhead since wireless links typically has low-bandwidth and nodes rely on batteries and usually have limited processing capacity and memory [7], [8]. Additionally, the concept of Ad-hoc networks makes them susceptible to attacks including modification, impersonation, and fabrication [9]. Accordingly, achieving an efficient and secure routing is a crucial concern in such networks [10]-[12].

In managed-open environment [13]-[15], such as that conducted by students in a campus, using already existed infrastructure is possible. So, there is a chance for pre-deployment of different keys and certificates to provide a starting point for promising security in such networks. However, depending on a single centralized server in Ad-hoc networks is not visible as this server is just a mobile node making it difficult for a node to communicate to that server. Additionally, the server may be the operation bottleneck as it is merely a normal limited-resources Ad-hoc node. Accordingly, the certificate authority must be distributed amongst multiple servers. Additionally, the importance of scalable protocols, together with the existence of small, inexpensive and low-power positioning tools rationalize utilizing position-based routing approaches in mobile Ad-hoc networks [9], [13], [14].

The abovementioned discussion, encourages us to propose S-Octopus to provide a scalable and secure position-based routing solution for Ad-hoc networks. In this paper, we discuss our new protocol S-Octopus and compare it to protocols presented in [15] and [9]. The three protocols are compared concerning their security level, scalability, robustness, load distribution, used route discovery phase, appropriate network density, requiring centralized trust and/or synchronization, implementation complexity, packet and processing overhead, route acquisition latency and data packets end-to-end delay.

Rest of the paper is arranged as follows. Section II presents some related works on Ad-Hoc routing protocols. Section III proposes the details of S-Octopus routing protocol. Section IV and Section V include a qualitative analysis and comparison of ARAN, ARANz and S-Octopus protocols. Our work is concluded in Section VI. Lastly, we highlight our future directions in Section VII.

II. Related Works

Related and recent researches on Ad-hoc routing protocols are discussed in this section. Subsections B and C present two protocols of interest; ARAN and ARANz.

A. Existing Works

Mobile Ad-hoc networks routing protocols are generally classified into topology-based and position-based. Topology-based protocols forward packets based on information about existing links between nodes. They are classified into proactive, reactive, and hybrid protocols. Proactive routing periodically broadcasts control messages in a try to help nodes always know a

Manuscript received June 11, 2021; revised August 12, 2021; accepted September 23, 2021.
Corresponding author: Liana Khamis Qabajeh (email: liana_tamimi @ppu.edu)

fresh route to other nodes [16]. It is obvious that proactive routing is not suitable for Ad-hoc networks since it continuously consumes nodes power [17], [18].

On the contrary, reactive routing initiates a route discovery procedure only upon having data packets to be sent and no periodic packets are mandatory. Numerous reactive protocols have been proposed including AODV protocol. Another example is Source-Initiated Link Expiration Time protocol (SILET) [19], which is a source-initiated reactive protocol that considers the predicted link expiration time while calculating the links weights. The destination selects the route having the minimum sum of the links weights. Reactive protocols may result in high control overhead in high-mobility networks and heavy-load situations. Scalability is considered as another weakness as they depend on blind broadcasts to determine routes. Hybrid protocols, as Zone Routing Protocol (ZRP) [18], seek to combine best properties of both proactive and reactive methods [20]. ZRP weakness is that it may perform as a pure proactive protocol for large routing zone, whereas for small zones it performs as a reactive one. Authors in [21] proposed a new hybrid routing protocol that utilizes advantages of both proactive and reactive approaches via allowing mobile nodes to flexibly run either a proactive or a reactive protocol considering their velocity and traffic. Topology-based approaches generally do not scale in networks having more than several hundred nodes [22]. Moreover, the aforementioned protocols inherently trust all participants. Apparently, this may result in security issues and allow routing attacks [15], [23].

After that, many efforts have been done on securing routing protocols including ARAN [15] and Secure Ad-hoc On-demand Distance Vector (SAODV) [24]. A specific protocol of interest is ARAN [15] protocol. ARAN is identical to AODV, but provides authentication of route instantiation and maintenance. ARAN aims to detect attacks by malicious nodes in managed-open networks where prior security coordination and trusted certificate authority server are required. Compared to original AODV, ARAN thwarts numerous attacks including modification, impersonation and fabrication. Even though ARAN has comparable performance to AODV, it results in extra packet overhead and latency due to signing packets. Moreover, ARAN depends on a centralized trust and has single point of failure and compromised server problems. ARAN has a scalability problem as any route request is flooded to the entire network.

Subsequently, position-based routing showed enhanced scalability and performance [22], [25], [26]. It utilizes nodes geographical positions to make routing decisions and improve performance and efficiency. Such protocols require each node to attain its own position and the destination position through Global Positioning System (GPS) and location service, respectively. Position-based protocols are classified into Restricted directional flooding, Greedy and hierarchical.

Most position-based protocols, as Greedy Perimeter Stateless Routing (GPSR) [27], use greedy forwarding to forward packets from source to destination; i.e., the source chooses, as the subsequent hop, a neighboring node that has the least cost towards the destination. In the same way, intermediate nodes choice their successor nodes until reaching the destination. Hence, nodes periodically send small beacons to announce their positions to allow other nodes to keep a one-hop neighbor table. These protocols are considered scalable ones since they do not require routes discovery and maintenance. However, periodic beacons cause network congestion and consume nodes energy. Furthermore, greedy forwarding is generally not guaranteed to find the ideal path. For instance, GPSR has good performance in dense environments; however, the situation becomes worse in sparse networks due to empty areas [22], [27].

Location-Aided Routing (LAR) [28] is an instance of restricted directional flooding routing in which source broadcasts packets towards all one-hop neighbors in the direction of the destination. In LAR, each node upon receiving a route request, compares its distance to the destination, with the distance of the preceding node to the destination. If the receiving node is nearer to the destination, it resends the packet; else, the node drops the packet. To discover the optimal path, numerous nodes are nominated for handling the route request packet and each one puts its IP address within the packet header. Saving the routes in the message header results in increasing the message size.

In hierarchical routing a multi-level hierarchy is utilized. In Terminal nodes framework (Terminodes) [29] for instance, if the number of hops between source and destination, packets are sent using a proactive distance vector. On the other hand, in long distance routing, greedy routing is utilized.

The above-mentioned position-based protocols are susceptible to several attacks as they do not consider security issues [23]. Moreover, many of them have little chance of finding the optimal route.

After that, many secure position-based routing proposals have been suggested including Anonymous On-Demand Position-based Routing in Mobile Ad-hoc Networks (AODPR) [23], Secure Geographic Forwarding (SGF) [30] and ARANz [9]. However, they still have many issues; including, single point of attack, high packet and processing overhead, and low scalability. ARANz, for example, intends to achieve security, attain robustness and alleviate the single point of attack problem by selecting numerous local certificate authorities. Furthermore, through dealing with the area as square-shaped zones and utilizing restricted directional flooding, it attains high-level of scalability and performance. However, assigning four LCAs on the boundaries of each zone result in high control overhead and latency in the communication among LCAs within a particular zone.

Recently, many researches have considered Ad-hoc networks security. Authors in [31]-[33], provided a detailed analysis of Ad-hoc networks security issues and the defeating proposals against different security attacks. A secured clustering technique based on a cryptography scheme for the urban area has been proposed in [34].

While a strong secure anonymous location-based routing method for MANET has been proposed in [35].

Other works suggested new security proposals to circumvent certain attacks. In [2], [36], [37], for example, novel flooding attack prevention protocols have been proposed. Authors in [7] and [38] suggested solutions for wormhole attack. While [10] and [11] considered black hole and grey hole attacks. Other works concentrated on proposing a new routing algorithm based on trust models or the reputation of the nodes depending on both direct and indirect trust to calculate a nodes trust value, in a try to allow only trusted nodes to participate in the chosen routes. These works include [39]-[43]. Authors in [31], [32], [44]-[46] presented detailed surveys of recent work conducted to secure Ad-hoc networks.

To summarize, various topology-based routing protocols yet have security vulnerabilities and are unscalable. Although, some security improvements have been proposed as in ARAN, the centralized node trust has raised other security and scalability difficulties. Lastly, restricted directional flooding achieves better performance compared to topology-based and other position-based protocols. ARANz employs restricted directional flooding, yet, introducing multiple local servers and maintaining network structure result in extra control overhead.

B. ARAN Protocol Overview

Authenticated Routing for Ad-hoc Networks (ARAN) [15] protocol ensures authentication of route discovery, setup, and maintenance. The main aim of ARAN is to defend against attacks from misbehaving nodes in a managed-open environment, hence, it expects a former security management. It involves a trusted Certificate Authority (CA) server whose public key is recognized by all other nodes. Before being able to participate in the network, every node should request a certificate from the CA. ARAN applies cryptographic certificates to avoid most security attacks against routing protocols. ARAN assures authentication, message integrity and non-repudiation for the Ad-hoc environment.

In ARAN, route discovery is conducted via propagating a Route Discovery Packet (RDP) from a source, and unicasting a REPly Packet (REP) from the destination through the opposite path to the source. Routing packets are authenticated at all hops from source to destination, and vice versa. Thus, every node upon forwarding a request or a reply signs it to enable the successor node to check its legitimacy. Since only the destination can issue REPs, loop freedom is assured. Nodes in ARAN retain one routing table entry per active source-destination pair, which is more expensive compared to per-destination entry in unsecure protocols.

ARAN prevents numerous attacks, such as spoofing, alteration and replay of routing messages. However, in addition to its scalability problem with the number of nodes, it introduces packet overhead and latency in route discovery as every packet should be signed. Lastly, ARAN depends on single certificate server which require keeping this server protected.

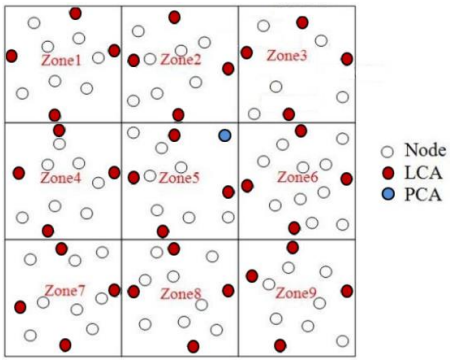

Fig. 1. ARANz network structure.

C. ARANz Protocol Overview

Like ARAN, ARANz implements cryptographic certificates, however, ARANz introduces a hierarchal distributed routing algorithm aiming to enhance performance and distribute load by dealing with the area as square-shaped zones. Furthermore, it aims to accomplish high-level of security and robustness, along with getting rid of the single point of failure and attack problems by allotting trust to several Local Certificate Authority (LCA) servers. Each zone has four LCAs that must work together to issue certificates to the nodes inside their zone. ARANz network structure is shown in Fig. 1, supposing that the network is divided into nine zones. Primary Certificate Authority (PCA) is a previously chosen node that has the needed software to divide the area into zones and select the initial LCAs.

Additionally, ARANz shows improved scalability, performance, and robustness against nodes movements via utilizing the restricted directional flooding position-based routing. When a node wants to send data to another, the source obtains the destination position from its zone LCAs. After that, restricted directional flooding is used to send the route request packet in an attempt to reduce overall overhead and save network bandwidth, compared to original ARAN protocol. Hence, moving nodes inform LCAs of their zones about their new locations. ARANz starts by network setup phase which consists of certifying nodes, dividing area into zones and picking preliminary servers. Network maintenance phase ensures preserving the constructed network structure considering some issues like refreshing certificates, LCAs synchronization, as well as nodes movement.

Location service phase allows the source to be aware of the destination position through communicating LCAs in its zone. After knowing the destination position, the route instantiation and maintenance phase is initiated. The source starts route discovery through sending a Route Discovery Packet (RDP) using restricted directional flooding towards the destination. When receiving the first RDP, destination sends a Route REPly (RREP) packet along the reverse path towards the source. After accomplishing route discovery and setup, the source starts forwarding data packets to the destination. To preserve the nominated path, nodes track whether routes are active or not and send ERRor (ERR) packets to report broken links within active routes.

The sender of any packet as well as intermediate nodes sign the packet using their private keys and append their certificates to the packets; to enable other nodes to validate signature using the public key which is extracted from the attached certificate. Accordingly, as in ARAN, the exchanged data packets among nodes are not signed and do not have appended certificates.

Dividing the network into multiple square-shaped zones and assigning four LCAs on the boundaries of each zone, make communication among LCAs of adjacent zones easier and faster on one hand, but result on high control overhead and latency in performing the operations that require communication among LCAs of the same zone such as updating nodes certificates, updating nodes positions, obtaining destination position, LCAs synchronization, and announcing malicious or compromised nodes.

III. S-Octopus Protocol

Our newly proposed routing protocol, S-Octopus, like ARAN and ARANz, uses cryptographic certificates to avoid most security attacks against Ad-hoc routing protocols. S-Octopus divides the area into sectors in a try to enhance performance and distribute load. Furthermore, it seeks to improve robustness and security, solve the single point of failure and attack problems through distributing trust amongst multiple Sector Certificate Authority (SCA) servers. SCAs are chosen to be closer to the network center to reduce the resulted overhead and latency from the communication among them. SCAs are arranged as a series, and adjacent SCAs should work together to distribute nodes certificates. S-Octopus also proposes a misbehavior detection scheme to discover the malicious and compromised nodes. Additionally, S-Octopus uses restricted directional flooding to send route request packets towards destinations in a try to improve scalability, reduce overhead and save network bandwidth. Thus, the SCAs act also as position servers and keep updated information about positions of nodes within their sectors.

TABLE I: VARIABLES AND NOTATIONS OF S-OCTOPUS

Notation	Description
N	Number of participating nodes
S	Number of sector-shaped regions
L	Network area length
W	Network area width
PCA	Primary Certificate Authority
SCA_{Ss}	Sector Certificate Authority of sector s
S_s	Sector number s
CK	Common Key
K_{NET-}	Network private key
K_{NET+}	Network public key
K_{Ss-}	Private key of sector s
K_{Ss+}	Public key of sector s
$Cert_{Ss}$	Sector s SCAs Certificate
K_{Nn-}	Private key of Node n
K_{Nn+}	Public key of Node n
$Cert_{Nn}$	Node n Certificate
IP_n	Node n IP address
N_n	Nonce issued by Node n
P_n	Node n Position
Pr_n	Probability of Node n to be chosen as SCA
t	timestamp of certificate creation
e	certificate expire time
Dc_n	Distance from Node n to area center point
S_n	Node n movement speed
B_n	Node n battery life time
C_n	Node n CPU power
M_n	Node n memory capacity
Dc_{max}	Maximum distance between a node and center point
S_{max}	Maximum probable node movement speed
B_{max}	Maximum probable battery life time
C_{max}	Maximum probable CPU power
M_{max}	Maximum probable memory capacity
X_{cp}, Y_{cp}	Network area center point coordinates
X_{co}, Y_{co}	Network area corner coordinates
d_{mov}	Pre-defined distance that a node is allowed to move from its last identified position before it must send its new position to its SCA
d_{cen}	Pre-defined distance that a SCA is allowed to move from the network center before it should initiate a new SCA election
W_1, W_2, W_3, W_4 and W_5	Weights of the considered parameters upon electing SCAs

TABLE II: PACKET IDENTIFIERS OF S-OCTOPUS

S-Octopus Stage	Packet identifier	Stands for
Network setup	NetSet	Network Setup
	NodeInfo	Node Information
	NodeRole	Node Role
Network maintenance	CertReq	Certificate Request
	CertRep	Certificate Reply
	AdjCertReq	Adjacent Certificate Request
	AdjCertRep	Adjacent Certificate Reply
	DepNode	Departed Node
	NewSector	New Sector
	NewNode	New Node
	SCAElection	SCA Election
	NewSCA	New SCA
	NewAdjSCA	New Adjacent SCA
	FailNode	Failed Node
	SysClock	System Clock
Location service	PosReq	Position Request
	PosRep	Position Reply
	AdjPosReq	Adjacent Position Request
	AdjPosRep	Adjacent Position Reply
Route instantiation and maintenance	RouteReq	Route Request
	RouteRep	Route Reply
	Error	Error
Data transmission	Data	Data
Misbehavior detection system	MisNode	Misbehavior Node
	ComNode	Compromised Node

Algorithm I shows the pseudocode for S-Octopus protocol. S-Octopus consists of six stages that are network setup, network maintenance, location service, route instantiation and maintenance, data transmission, and lastly misbehavior detection system. The details of different S-Octopus stages are presented in the following subsections. Table I displays variables and notations used with S-Octopus protocol whereas Table II summarizes the used packet identifiers.

A. Network Setup

S-Octopus assumes N nodes in a managed-open environment, which are distributed randomly in $L \times W$ km^2 network and are aware of their positions. This network will be partitioned into S sector-shape regions. Primary Certificate Authority (PCA) is a previously selected node that owns the private part of the network key (K_{NET-}) and contains the software required to initiate the network setup, split the area into sectors and select the initial SCAs. All participating nodes own a private/public key

pair, the network key public part (K_{NET+}) and a Common Key (CK) that is used for encrypting and decrypting packets sent by non-PCA nodes during network setup stage.

```
Algorithm I: Pseudocode for S-Octopus protocol

Start
Network setup stage: Dividing area into sectors, electing SCAs and issuing certificates
While not end of network lifetime
  {Network maintenance stage: Considering nodes movement, nodes failure, updating certificates and SCAs synchronization
   Misbehavior detection system: Reporting misbehaving nodes via MisNode and ComNode packets

   If any source has data to be sent
     {If there is no established route between source and destination
        {If source is not aware of the destination position
           Location service stage: Source obtains destination position via SCA of its sector
         Else
           Route instantiation:
              Route discovery: Sending a RouteReq using restricted directional flooding towards destination
              Route setup: Sending RouteRep via reverse path
        }// If there are no established routes between source and destinations
      Else
        If there is any broken link in the established route
           Route maintenance: Sending Error packet due to link breakage
        Else
           Data transmission stage: Source forwards Data packets towards destination through the established route
      }// If any source has data to be sent

  }// While not end of network lifetime
End
```

The PCA initiates network setup with broadcasting a packet informing the nodes about starting the Network Setup (NetSet). The packet is signed with K_{NET-} as an evidence that the PCA is really the source of the packet. Upon receiving the first NetSet packet, nodes record the previous node IP address, continue sending the packet and respond by sending a Node Information (NodeInfo) packet to the PCA. NodeInfo contains the IP address of the node (IP_n) and the required information to select the SCAs. These messages are encrypted with the CK. Upon getting a NodeInfo packet, nodes try to decrypt it via CK to guarantee that the previous node is trusted and to continue processing the packet; else node drops the packet. Following to encrypting the NodeInfo packet, it is forwarded through the reverse route until it reaches PCA.

Subsequent to receiving the NodeInfo packets from all certified nodes, PCA divides the network into several sectors and assigns a SCA for each sector. Fig. 2 shows the network structure, supposing that the network is divided, for instance, into eight sectors (number of octopus legs). While Fig. 3 shows the arrangement of SCAs as an anti-clockwise ring, which will be used later during the network maintenance, location service, and misbehavior detection system stages. Algorithm II shows the pseudocode for determining the sector that each node belongs to.

Upon electing SCAs, each Node n within a sector S_s is given a weight indicating its probability to be elected as the SCA of that particular sector. Some important points that are considered upon choosing SCAs are the distance between the node and the center point of the network area (Dc_n), speed of the node (S_n) and remaining life time of the node battery (B_n). Selecting a SCA that is near the center point of the network area and having a low speed increases the chance that the communication between SCAs of different sectors will be conducted using single hop, hence, guarding critical packets and reducing overhead. Selecting SCAs with low movement speed also increases the period that the SCA stays in the sector and so delays the need to re-elect a new SCA. Furthermore, selecting a node having high battery remaining life time decreases the likelihood of having its battery off, i.e., reduces the probability of electing a new SCA and transferring important and secure information it possesses.

CPU processing power (C_n) and memory (M_n) of the nodes are also significant aspects upon selecting SCAs. Having high CPU processing power and satisfactory memory highly affect performance as these SCAs are the operation bottleneck of the position management system.

```
Algorithm II: Pseudocode for determining
              the current sector of a node

Start
   Node n position (Xn, Yn)
   Network area center point (Xcp, Ycp)
   θ = 45°
   xResult = Xn - Xcp
   yResult = Yn - Ycp
   β = tan-1 (Yn - Ycp / Xn - Xcp)

   If xResult > 0 and yResult > 0
        {If β <= θ          Sector = 1
         Else               Sector = 2}
   Else If xResult < 0 and yResult > 0
        {If β <= θ          Sector = 3
         Else               Sector = 4}
   Else If xResult < 0 and yResult < 0
        {If β <= θ          Sector = 5
         Else               Sector = 6}
   Else If xResult > 0 and yResult < 0
        {If β <= θ          Sector = 7
         Else               Sector = 8}
End
```

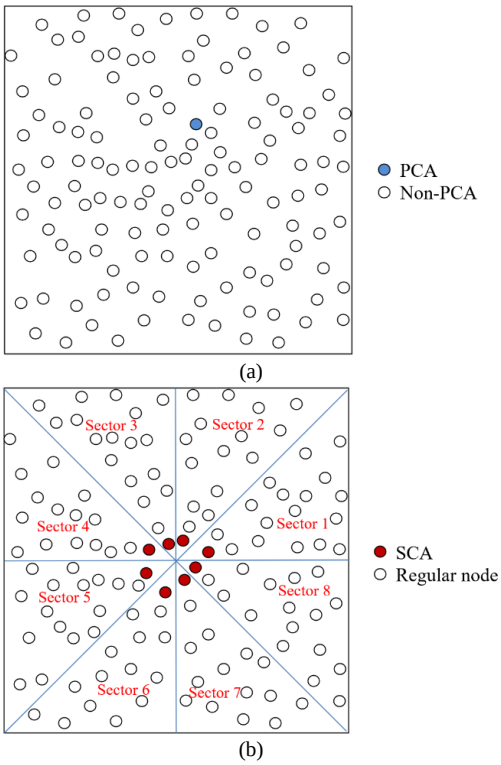

Fig. 2. S-Octopus network structure

Fig. 3. Arrangement of SCAs as an anti-clockwise ring

The PCA uses the received NodeInfo packets to compute the possibility of different nodes inside a specific sector to be elected as a SCA for that sector. The probability (Pr_n) of Node n to be elected as a SCA of the sector where it resides is given as:

$$\mathrm{Pr}_n = W_1\left(1-\frac{\mathrm{Dc}_n}{\mathrm{Dc}_{\max}}\right)+W_2\left(1-\frac{S_n}{S_{\max}}\right)+ W_3\frac{B_n}{B_{\max}}+W_4\frac{C_n}{C_{\max}}+W_5\frac{M_n}{M_{\max}}$$

where $\mathrm{Dc}_{\max}$ is the maximum probable distance from a node to the center point of the network area, $S_{\max}$ is the maximum probable node movement speed, $B_{\max}$ is the maximum probable battery life time, $C_{\max}$ is the maximum probable CPU power, and $M_{\max}$ is the maximum probable memory capacity, and W_1, W_2, W_3, W_4 and W_5 are the weights of the considered parameters upon electing SCAs.

Distance Dc_n from node position $P_n = (X_n, Y_n)$ to center point $(X_{\mathrm{cp}}, Y_{\mathrm{cp}})$ of network area is given as:

$$\mathrm{Dc}_n = \sqrt{(X_n - X_{\mathrm{cp}})^2 + (Y_n - Y_{\mathrm{cp}})^2}$$

$\mathrm{Dc}_{\max}$ is calculated once as distance between the center point of network area and one of the network corners. Referring to Fig. 4, $\mathrm{Dc}_{\max}$ may be evaluated as the distance between the center point $(X_{\mathrm{cp}}, Y_{\mathrm{cp}})$ of network area and one of the area corners, $(X_{\mathrm{co}}, Y_{\mathrm{co}})$ for example.

$$\mathrm{Dc}_{\max} = \sqrt{(X_{\mathrm{co}} - X_{\mathrm{cp}})^2 + (Y_{\mathrm{co}} - Y_{\mathrm{cp}})^2}$$

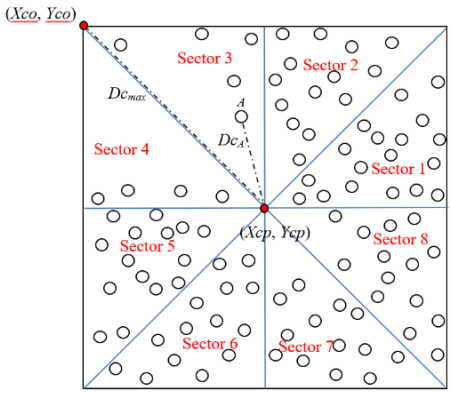

Fig. 4. The maximum probable distance from any node to the center point of the network area

Subsequently, PCA unicasts a Node Role (NodeRole) packet to every participating node to enable it to know its role in the network (SCA or regular node). These packets are sent using source routing as the PCA is aware of all nodes positions. Regular Node n, for instance, will receive a unicast NodeRole message containing node certificate (Cert_{Nn}), sector number where it is located (S_s), identity and position of SCA in its sector (SCA_{Ss}), and its sector public key (K_{Ss+}). Node n certificate (Cert_{Nn}) contains its IP address (IP_n), its public key (K_{Nn+}), timestamp (t) of certificate creation time, and certificate expire time (e). These certificates are signed by the $K_{\mathrm{NET}-}$ and are used by nodes to authenticate themselves while exchanging network maintenance, position and routing messages.

A NodeRole message is also unicasted for each SCA including node certificate, sector SCA certificate (Cert_{Ss}), sector private/public key pair, identity and position of SCA of immediate adjacent sector (successor SCA in the ring), and public key of the immediate adjacent sector. Message also includes authentication table that contains information about nodes inside this sector such as IP address, position, public key, certificate creation time, and certificate expiration time. This table is used upon updating nodes certificates. As well, it is used by SCAs when getting a position request packet to know if the destination is external or local; in order to issue position request packet to successor SCA or respond with a position reply to the source respectively.

Sector SCA certificate (Cert_{Ss}) is used by a SCA as a proof that it is a SCA of a specified sector. Cert_{Ss} includes the sector number, sector public key, certificate creation time, and certificate expiration time. The sector private key is used to sign these certificates. These certificates are used between successor SCAs and between SCAs and nodes in their sectors while exchanging network maintenance and position messages.

B. Network Maintenance

Following setting up the network stage, nodes can update their certificates as well as moving freely within the network. S-Octopus handles these issues. Nodes use certificates to assure authentication. So, the source node signs the packet via its private key and attaches its node certificate. If the source node is a SCA, it also includes sector SCA certificate to allow the destination to ensure

that the SCA has a valid certificate for that precise sector. Nodes along the way validate previous node signature, remove certificate and signature of the previous node, sign the original contents of the packet, and add their own certificates.

Restricted directional flooding is used to send packets from nodes to SCA of their sector since they are aware of the position of this SCA. Restricted directional flooding is also used for communication between nodes after acquiring the destination position by the source. Similarly, communications between adjacent SCAs in neighboring sectors is done using restricted directional flooding; if they are unreachable within one hop. On the other hand, sending packets from SCAs to nodes in their sectors is done using source routing; since positions of nodes are known by SCAs of their sectors.

Lastly, reply packets are directed over reverse paths of their related request packets. It is left as implementation option to take into consideration high-mobility nodes in dynamic networks and regions without nodes in sparse networks. Hence, if a source node did not receive a reply packet after 3 attempts, for example, the request packet is sent to the entire network.

1) Certifications update

Every node must keep valid certificate with the SCA of its sector. This is achieved by periodic Certificate Request (CertReq) packets sent from nodes to SCAs. These CertReq packets are signed by the nodes private key and sent using restricted directional flooding. Fig. 5, shows the certificate request packets sent for updating Node A certificate.

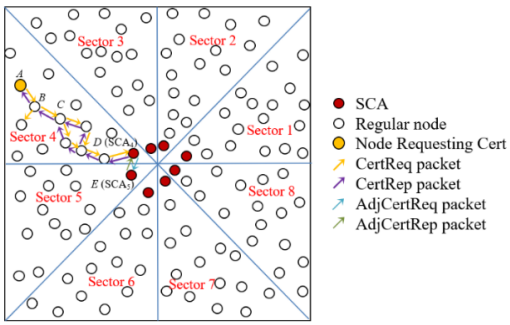

Fig. 5. Node A certification update

Intermediate nodes, upon receiving a CertReq packet, set up a reverse route towards the source by storing the neighbor from which they received the packet. A receiving node uses public key of its preceding one to authorize the signature and prove that the certificate is not expired. This node also ensures that it has not yet processed this CertReq. It also signs the packet, attaches its own certificate, and continue sending the packet.

Upon receiving the first CREQ, the corresponding SCA packet communicates its successor SCA asking it to update this certificate or not. This is accomplished by issuing Adjacent Certificate Request (AdjCertReq) packet. Thus, SCAs are allowed to give a certificate only if they receive Adjacent Certificate Reply (AdjCertRep) packet from their successor SCA. This helps in enhancing security; an adversary will not be able to issue certificates to untrusted nodes unless it compromised two adjacent SCAs. Then the SCA will send a Certificate Reply (CertRep) packet through the reverse path towards the source.

SCAs also should maintain fresh node and sector SCA certificates. So, each SCA should periodically unicast AdjCertReq to its successor SCA. This SCA is issued both node and sector SCA certificates upon receiving the AdjCertRep.

2) Nodes mobility

To enable SCAs to track nodes movement inside a sector and among sectors, regular nodes must include their new position in the CertReq packet sent to their SCA upon moving a pre-defined distance (d_{mov}) from their last known location.

Upon leaving its current sector, the original sector SCA of the leaving node removes node information from its table and sends a Departed Node (DepNode) packet to its adjacent SCA. DepNode packet assures that the departing node is reliable and includes its position. Each SCA in turn, upon receiving this packet will forward it to the successor SCA until the packet reaches SCA of the intended sector. Fig. 6 shows messages sent when A departs sector number 4 to sector number 7 (leaves position P_A towards P'_A). The new sector SCA sends a New Sector (NewSector) packet to the moving node; holding the new sector number and public key, along with IP address and position of SCA of the new sector. This SCA also sends New Node (NewNode) packet to successor SCA informing it about the arriving node.

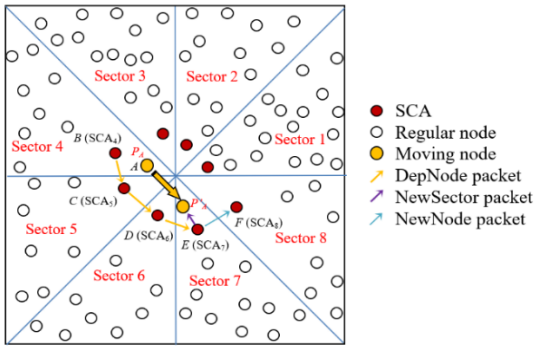

Fig. 6. Movement of node A from sector 4 to sector 7

Upon moving pre-defined distance (d_{mov}) from its last known position, SCA broadcasts its new position to nodes within its sector. It also sends its position to SCA of adjacent sector. Moreover, the distance from SCA location to the network center may become larger than a pre-defined distance (d_{cen}), or SCA may choose to leave its sector. So, a new SCA election is mandatory. Upon deciding to depart its sector, SCA sends a SCA Election (SCAElection) packet to nodes residing currently within its sector. Nodes within this sector calculate and send their probability over reverse path to the leaving SCA. Then, the leaving SCA choses the node with the maximum probability to be the new SCA. So, it sends a New SCA (NewSCA) packet to nodes in this sector to be aware of the address and position of the fresh SCA. This information is also sent to the successor SCA in the adjacent sector through a New Adjacent SCA (NewAdjSCA) message. Then, the old SCA handovers to the new SCA the needed information.

3) Nodes failure

It is left as implementation option to take into consideration nodes failure. In this case, a SCA backup strategy should be implemented in the system. To maintain acceptable level of security, each SCA should periodically send a copy of half the information it contains to its successor SCA and the other half to its precedence one.

The sudden failure of an SCA is discovered from the periodic sector certificate update of SCAs. So, if the SCAs in a particular sector did not receive the AdjCertReq packet from a particular SCA within a pre-determined time it discovers that this SCA has a problem. Then, successor and precedence SCAs take the duty of selecting a new SCA and sending NewSCA packet. To allow the failed SCA to rejoin the network from any sector after repairing, node IP address and public key are sent to all SCAs in the network. So, every SCA sends a Failed Node (FailNode) packet to its successor SCA.

Periodic node certificate update helps in discovering regular nodes failure. If the authentication table of a SCA contains an expired node certificate, and no CertReq packet has been received within a pre-defined period of time, SCA concludes that this node is corrupted. So, the SCA that issued the last certificate to the node sends a FailNode packet.

4) SCAs synchronization

SCAs must maintain synchronized clocks, for instance, to circumvent issuing certificates with two different time stamps to two nodes in different sectors at the same moment. Hence, nodes have their local clocks running independently, while tracking the gap between their clocks and the system clock.

As a preliminary step, PCA includes a time stamp inside the NewRole message forwarded to SCAs during the network setup stage. So SCAs will know the variation between its local clock and the PCA clock. Moreover, a time stamp is incorporated in the information sent to newly elected SCAs.

Additionally, clocks may have drifts, and oscillator's frequency may vary unpredictably due to a variety of physical effects [9]. So, periodically, one of the SCAs sends a System Clock (SysClock) message including a time stamp to other sectors SCAs to reduce SCAs clocks drifts. To raise robustness and distribute load, the SCAs takes turn to send this message considering the anti-clockwise SCAs ring arrangement. Additionally, replay attacks are avoided using a nonce. Definitely, SCAs include their sector SCA certificates within the packet, sign the packet contents, and append their own certificates.

The timestamp included in their certificates, is used by regular nodes to be aware of the system time and check other nodes certificates validity; hence, no more communications between SCA and regular nodes in a specific sector are needed.

C. Location Service

Prior to starting route discovery process, the source is supposed to obtain the position of the destination. Referring to Fig. 7, source *S* sends a Position Request (PosReq) packet to its sector SCA using restricted directional flooding to enquiry about the location of the destination D.

When the SCA receives the first PosReq, it checks if the destination is local or not. If the source and the destination are within the same sector, the destination is found in the SCA authentication table. So, the SCA unicasts a Position Reply (PosRep) packet towards the source. This PosRep includes the position of the destination and passes through the reverse way towards the source.

In case that the destination is not in the same sector, the SCA sends Adjacent Position Request (AdjPosReq) to its successor SCA in the adjacent sector. Forwarding the AdjPosReq packet among SCAs in the ring continues till finding the destination in the authentication table of one of the SCA_s (SCA_6 in Fig. 7). SCA_6, consecutively, unicasts an Adjacent Position Reply (AdjPosRep) back to source sector SCA. Source SCA unicasts a PosRep along the reverse way to source. Position discovery packets are authenticated by each hop.

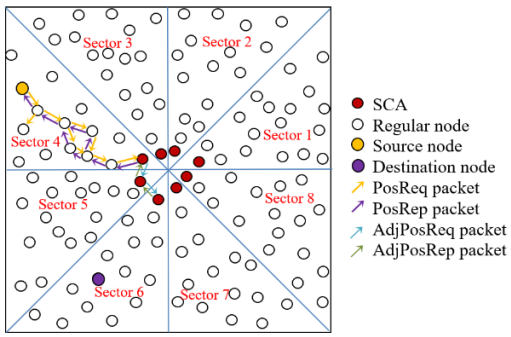

Fig. 7. Authenticated location service.

D. Route Instantiation and Maintenance

Upon obtaining the destination position, the source starts route instantiation via sending a Route Request (RouteReq) packet using restricted directional flooding to the neighbors of the source. Upon receiving the first RouteReq, the destination unicasts a Route Reply (RouteRep) packet back to the source. All route instantiation steps are authenticated hop-by-hop.

If no data is received during a route lifetime, this route is disabled. Moreover, data received on a disabled route results in generating an Error (Error) packet. Error packets are also used to announce broken links in active routes. Like ARAN and ARANz, S-Octopus utilizes neighbors hello packets to identify links failures.

E. Data Transmission

After instantiating the route, the source starts forwarding the data to the destination. As in ARAN and ARANz, once the route reply reaches the source, it is assured that the selected route is authentic. As a result, the exchanged Data (Data) packets after setting up a route do not have appended certificates and are not signed.

F. Misbehavior Detection System

Upon noticing misbehaving of other nodes, regular node sends Misbehavior Node (MisNode) packet to report this to its sector SCA. Upon receiving a pre-defined number of MisNode packets for the same node by

the majority of the SCAs, they collaborate and broadcast a Compromised Node (ComNode) packet. Hence, nodes exclude misbehaving node from the routing activities till its certificate expires normally.

Similar technique is used if SCAs detect misbehavior of a particular SCA in the network; e.g., if four SCAs have detected that a specific SCA has misbehavior actions, they remove this SCA from the SCAs ring, broadcast a ComNode packet and conduct a new SCA election procedure. S-Octopus responds to all packets having erratic behaviors by dropping them. Erratic behaviors include using invalid certificates, inappropriately signed packets, and misuse of some packets.

IV. Comparison among Discussed Protocols

A summary of the studied protocols along with the considered evaluation criteria are presented in Table III. This study is a high-level qualitative discussion rather than a precise quantitative performance evaluation of the protocols. ARAN is a reactive topology-based routing protocol that conducts route discovery process using broadcasting; while ARANz and S-Octopus are position-based restricted directional flooding approaches. The availability of inexpensive and low-power positioning instruments justifies adopting position-based routing in ARANz and S-Octopus protocols.

Table III: Features of the Studied Protocols

Criterion	Protocol		
	ARAN	**ARANz**	**S-Octopus**
Approach	Topology-based (Reactive- Broadcasting)	Position-based (Restricted directional flooding)	Position-based (Restricted directional flooding)
Secure improvement on	AODV	AODV	AODV
Security techniques	Timestamps and Certificates	Timestamps and Certificates	Timestamps and Certificates
Main idea/contribution	Protect routing packets against misbehaving nodes attacks within managed-open environments.	Solve scalability and single point of failure and compromise problems found in ARAN.	• Solve scalability and single point of failure and compromise problems found in ARAN. • Reduce the required time and control overhead associated with communication among different SCAs.
Proposal	• Assures route discovery, setup and maintenance authentication. • Prevents most Ad-Hoc routing protocols security attacks. • Assures authentication at every hop from source to destination, and from destination to source.	• Divides network into zones and elects many LCAs in each zone. • Involves issuing a PDP if the destination position is unknown to the source. • Prevents most Ad-Hoc routing protocols security attacks. • Assures authentication at every hop from source to destination, and from destination to source.	• Divides network into sectors and elects many SCAs in the network. • Involves issuing a PosReq if the destination position is unknown to the source. • Prevents most Ad-Hoc routing protocols security attacks. • Assures authentication at every hop from source to destination, and from destination to source.
Loop Freedom	Yes	Yes	Yes
Density	All	All	All
Path Selection	Quickest	Quickest	Quickest
Central trust	Certificate Authority	No	No
Load Distribution	No	Yes	Yes
Synchronization	No	Yes	Yes
Robustness	Medium	High	High
Scalability	Low	Medium	High
Implementation Complexity	Medium	High	High
Packet Overhead	High	Medium	Low
Processing Overhead	Medium	Medium	Medium
Route Acquisition Latency	High	Medium	Low
Data Packets End-to-End delay	Medium	Medium	Medium
Advantages	Robust against many security attacks.	• Robust against many security attacks. • Moderate scalability. • No single point of failure and compromise, i.e., higher robustness and availability. • Moderate packet overhead.	• Robust against many security attacks. • High scalability. • No single point of failure and compromise, i.e., higher robustness and availability. • Reduced packet overhead.
Disadvantages	• Single point of failure and compromise, and so reduced robustness and availability. • Scalability problem regarding number of nodes due to blind broadcasts to discover routes.	• LCAs synchronization. • Additional hardware (GPS). • Further delay to inquiry about the position of the destination.	• SCAs synchronization. • Additional hardware (GPS). • Further delay to inquiry about the position of the destination.

The three presented protocols use cryptographic certificates to avoid most of the Ad-Hoc routing protocols attacks and guard against erratic behaviors. Hence, they achieve authentication, confidentiality, integrity, non-

repudiation, and anti-spoofing. Both ARANz and S-Octopus try to achieve a better security and circumvent single point of attack by allotting trusts to several CAs. The three protocols are suitable for any network density. Also, all of them are loop-free and preserve network resources and assure correct operation.

Data is forwarded through the quickest path taken by the route discovery packet that reaches the destination first. Experiments in [9] and [15] show that the average path length of AODV, ARAN and ARANz are almost equal; indicating that the first route discovery packet reaching the destination typically goes through the shortest path. Hence, it is expected for S-Octopus to be the same. Each node in ARAN ought to update its certificate by contacting the trusted CA server; hence the load is concentrated on that CA. CA also demonstrates a centralized trust and a single point of attack. ARANz and S-Octopus, however, try to distribute load and trust by dealing with the area as regions and announcing numerous CAs. Using multiple CAs, on the other hand, requires synchronizing them.

ARAN robustness in the route discovery stage is higher compared to ARANz and S-Octopus since it broadcasts the route request to the entire network. ARANz and S-Octopus, on the contrary, use restricted directional flooding to discover routes, increasing the single node failure and movement effect. After route setup, robustness of the three protocols is the same, since an individual node failure may result in losing a packet and setting up a new route. ARANz and S-Octopus try to attain enhanced robustness compared to ARAN by distributing trust amongst numerous CAs; multiple CAs collaborates to produce nodes certificates and act as backups of each other. However, upon the failure of the sole CA in ARAN, all the nodes will not be able to update their certificates. Considering the aforementioned points, the robustness of ARAN is considered medium and those of ARANz and S-Octopus are considered as high.

Scalability is concerned with the protocol performance upon increasing number of nodes within the network. ARAN assumes the presence of one CA, which comprises the operation bottleneck especially in large networks. Furthermore, increasing nodes in the network while using broadcast will increase the packet overhead due to broadcasting RREQ packets. Hence, ARAN scalability is considered to be low.

ARANz and S-Octopus are able to achieve higher scalability since they are capable of preserving good performance even in large networks. High performance is achieved via implementing restricted directional flooding instead of broadcasting, separating the network into regions and distributing load among multiple CAs. Location service packets are expected not to highly affect scalability since source routing or restricted directional flooding is used to send these packets. Even CA election process is performed by sending a packet to nodes in the intended region only. In ARANz however, dividing the network into several square-shaped zones and assigning four LCAs on the boundaries of each zone, result on larger number of LCAs and higher control overhead and latency in communications among them. The situation will be worse in large networks with large number of zones. In S-Octopus, however, SCAs are selected as close as possible to the center of the network which reduces the resulted overhead from SCAs communication. Accordingly, scalability of ARANz is considered to be medium and that of S-Octopus to be high.

Implementation complexity describes the difficulty of implementing and testing a specific protocol. This criterion is extremely subjective. ARAN implementation complexity is considered to be medium due to certification update and messages encryption/decryption. ARANz and S-Octopus have higher implementation complexity due to their security considerations, treating the network as regions and electing several CAs.

Packet overhead considers bandwidth consumption resulted from larger packets and/or larger number of sent packets. ARAN protocol has a high packet overhead due to the large-size packets as a result of signatures and certificates included within packets and higher number of packets as a result of broadcasting. ARANz packet overhead is considered to be medium due to the large-size packets as a result of the used security techniques and reduced number of packets compared to ARAN owing to using restricted directional flooding. Location service messages is not expected to significantly affect packet overhead, especially when the source and destination resides within the same zone; due to using source routing or restricted directional flooding.

Additionally, LCA election process and certificate updates are conducted within the anticipated zone. S-Octopus is expected to have lower packet overhead compared to ARANz due to reduced SCAs communication overhead. Processing overhead tackles processing requirements of each protocol. The three protocols have moderate processing overhead related to dealing with signatures and certificates.

The spent time from sending a route request packet by a source till receiving the first correlated route reply is referred to as route acquisition latency. Experiments in [15] confirm that the average route acquisition latency of ARAN is roughly double that for AODV owing to ARAN cryptographic operations. Simulations in [9] demonstrate that ARANz has reduced route acquisition latency compared to ARAN due to RDP broadcast in ARAN; i.e., processing RDP packets for other route detection processes by other nodes is delayed till this RDP is processed; thus, other routes acquisition latencies are increased. However, ARANz reduces number of processed packets by each node due to using restricted directional flooding. It is expected for S-Octopus to be able to achieve lower route acquisition latency compared to ARANz due to reduced SCAs communication overhead, which in turn results in decreasing waiting time required to process RDP packets by each node. Accordingly, route acquisition latency is expected to be high for ARAN, medium for ARANz, and low for S-Octopus.

Data packets End-to-end delay is the gap between sending a data packet by the source and its receipt at the

corresponding recipient. This comprises total delays for position discovery, route acquisition, intermediate nodes buffering and processing, and MAC layer retransmission. End-to-end delay of the three protocols is considered medium. Processing of data packets is almost the same for either protocol. Moreover, simulations in [9] and [15] show that data packets end-to-end delay for the AODV, ARAN and ARANz protocols are roughly equal. The route acquisition latency effect on data packets average end-to-end delay is not considerable; since the performed route discoveries number is small compared to the delivered data packets number.

V. Discussion and Analysis

The three presented protocols are efficient in discovering the optimal path at different network densities. They, utilize cryptographic certificates to guard against most Ad-Hoc routing attacks and have almost equal end-to-end data packets delay. ARAN has centralized trust and load, scalability problem due to certificate server operation bottleneck, and increased packet and processing overheads due to route request broadcast.

ARANz adopts ARAN authentication procedures. Through considering the network as zones and using restricted directional flooding, ARANz exhibits improved performance and scalability. ARANz attempts to distribute load and trust via introducing several LCAs within different zones. This aids in attaining enhanced security and robustness, and circumventing single point of attack and failure. Several LCAs in ARANz, on the other hand, rises a need to synchronize them. Moreover, partitioning the network into several square-shaped zones and appointing multiple LCAs on the boundaries of these zones, result on high control overhead and latency in accomplishing the operations that require communication among LCAs such as updating nodes certificates, updating nodes positions, obtaining destination position, LCAs synchronization, and announcing malicious or compromised nodes.

S-Octopus comes to overcome problems associated with both ARAN and ARANz protocols. S-Octopus avoids the single point of attack and failure problems associated with ARAN by introducing multiple SCAs. It also solves ARANz problems by dividing the area into sector-shaped regions and selecting several SCAs to be as close as possible to the center of the network. This will help S-Octopus to achieve higher scalability by reducing the resulted SCAs communication overhead, which in turn reduces overall packet overhead and packet processing latency.

VI. Conclusion

This paper has proposed a new routing protocol model; S-Octopus. S-Octopus targets the managed-open environment where already established infrastructure is assumed. S-Octopus presents a distributed routing that enhances performance and scalability via sector-shape area division. Introducing numerous SCAs helps in attaining robustness, enhancing security and mitigating the single point of failure and attack problems. S-Octopus also attempts to show improved scalability and robustness against continuous topological changes via restricted directional flooding. A qualitative comparison among ARAN, ARANz and S-Octopus protocols has been presented in this research.

VII. Future Works

Our next step is to use a simulation tool to evaluate and study the performance of our newly proposed protocol, S-Octopus. It is planned to evaluate S-Octopus effectiveness in achieving security requirements. Comparisons with other existing protocols including ARAN and ARANz will be performed. S-Octopus scalability in quite fast node mobility, large networks, and diverse percent of malicious nodes will also be tested.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] S. Singha, B. Jana, Sh. Jana and N. Mandal, "A survey to analyze routing algorithms for opportunistic network," *Procedia Computer Science*, vol. 171, pp. 2501–2511, 2020.

[2] L. Huy, L. Ngoc, and N. Tam, "AOMDV-OAM: A security routing protocol using OAM on mobile Ad hoc network," *Journal of Communications*, vol. 16, no. 3, pp. 104-110, 2021.

[3] F. Khan, A. Khan, S. Khan, I. Qasim, and A. Habib, "A secure core-assisted multicast routing protocol in mobile ad-hoc network," *Journal of Internet Technology*, vol. 21, no. 2, pp. 375-383, 2020.

[4] D. Ahmed and O. Khalifa, "An overview of MANETs: applications, characteristics, challenges and recent issues," *International Journal of Engineering and Advanced Technology*, vol. 6, no. 4, pp. 128-133, 2017.

[5] M. Divya, S. Subasree, and N. Sakthivel, "Performance analysis of efficient energy routing protocols in MANET," *Procedia Computer Science*, vol. 57, pp. 890-897, Dec. 2015.

[6] B. Mahalakshmi and S. Kumari, "An adaptive routing in flying Ad-hoc networks using FMCC protocol," *International Journal of Recent Technology and Engineering*, vol. 8, no. 5, pp. 2277-3878, 2020.

[7] J. Maxa, M. Mahmoud, and N. Larrieu, "Performance evaluation of a new secure routing protocol for UAV Ad hoc network," in *Proc. IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, 2019, pp. 1-10.

[8] A. Prasanth and S. Pavalarajan, "Implementation of efficient intra- and inter-zone routing for extending network consistency in wireless sensor networks," *Journal of Circuits, Systems and Computers*, vol. 29, no. 8, pp. 1-25, 2020.

[9] L. Qabajeh, M. Mat Kiah, and M. Qabajeh, "A more secure and scalable routing protocol for mobile ad hoc networks," *Security and Communication Networks*, vol. 6, no. 3, pp. 286-308, 2013.

[10] A. Tami, S. Hacene, and M. Cherif, "Detection and prevention of blackhole attack in the AOMDV routing protocol," *Journal of Communications Software and Systems*, vol. 17, no. 1, pp. 1-12, 2021.

[11] D. Kukreja, D. Sharma, S. Dhurandher, and B. Reddy, "GASER: genetic algorithm-based secure and energy aware routing protocol for sparse mobile ad hoc networks," *International Journal of Advanced Intelligence Paradigms*, vol. 13, no. 1-2, pp. 230-259, 2019.

[12] C. Perkins and E. Royer, "Ad hoc on-demand distance vector routing," in *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90-100.

[13] L. Qabajeh and M. Qabajeh, "Detailed security evaluation of ARANz, ARAN and AODV protocols," *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 5, pp. 176-192, 2020.

[14] L. Qabajeh and M. Qabajeh, "Detailed performance evaluation of ARANz, ARAN and AODV protocols," *Journal of Theoretical and Applied Information Technology*, vol. 98, no. 12, pp. 2109-2131, 2020.

[15] K. Sanzgiri, D. LaFlamme, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer, "Authenticated routing for Ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 598-610, 2005.

[16] A. Hassani, A. Sahel, and A. Badri, "FTC-OF: Forwarding traffic consciousness objective function for RPL routing protocol," *International Journal of Electrical and Electronic Engineering & Telecommunications*, vol. 10, pp. 168-175, March 2021.

[17] A. Aneiba and M. Melad, "Performance evaluation of AODV, DSR, OLSR, and GRP MANET routing protocols using OPNET," *International Journal of Future Computer and Communication*, vol. 5, no. 1, pp. 57-60, 2016.

[18] Z. Haas, M. Pearlman, and P. Samar, "The performance of query control schemes for the zone routing protocol," *ACM/IEEE Transactions on Networking*, vol. 9, no. 4, pp. 427-438, 2001.

[19] N. Meghanathan, "A unicast MANET routing protocol to simultaneously minimize the stability-hop count tradeoff and end-to-end delay," in *Proc. Ninth Int. Conf. on Information Technology - New Generations*, 2012, pp. 60-64.

[20] H. Toulni and B. Nsiri, "Hybrid routing protocol for VANET using ontology," *Procedia Computer Science*, vol. 73, pp. 94-101, October 2015.

[21] K. Soodl and N. Sah, "New unicast routing protocol using comparative study of proactive, reactive and hybrid protocols for MANET," *International Journal of Engineering Research and General Science*, vol.2, no. 4, pp. 44-50, 2014.

[22] Y. Cao and S. Xie, "A position based beaconless routing algorithm for mobile Ad hoc networks," in *Proc. Int. Conf. on Communications, Circuits and Systems*, 2005, vol. 1, pp. 303- 307.

[23] S. M. Rahman, M. Mambo, A. Inomata, and E. Okamoto, "An anonymous on- demand position-based routing in mobile Ad hoc networks," in *Proc. Int. Symposium on Applications and the Internet*, 2006, pp. 300-306.

[24] G. Zapata, "Secure Ad hoc on-demand distance vector routing," ACM *Mobile Computing and Communications Review (MC2R)*, vol. 6, no. 3, pp. 106-107, 2002.

[25] A. Guillen-Perez, A. Montoya, J. Sanchez-Aarnoutse, and M. Cano, "A comparative performance evaluation of routing protocols for flying Ad-hoc networks in real conditions," *Applied Sciences*, vol. 11, pp. 4363, May 2021.

[26] N. Gupta, R. Yadav, R. Nagaria, and D. Gupta, "An angular 3D path selection protocol in wireless sensor networks," *Open Computer Science*, vol. 11, no. 1, pp. 190–207, 2021.

[27] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. 6th Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking*, 2000, pp. 243-254.

[28] Y. Ko and N. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," *Wireless Network*, vol. 6, no. 4, pp. 307-321, 2000.

[29] L. Blazevic, L. Buttyan, S. Capkum, S. Giordano, J. Hubaux, and J. Le Boudec, "Self- organization in mobile ad-hoc networks: The approach of terminodes," *IEEE Communication Magazine*, vol. 39, no. 6, pp. 166-174, 2001.

[30] J. Song, V. Wong, and V. Leung, "Secure position-based routing protocol for mobile ad hoc networks," *Elsevier Ad Hoc Networks Journal*, vol. 5, no. 1, pp. 76-86, Elsevier, 2007.

[31] S. Thapara and S. Sharmab, "Attacks and security issues of mobile Ad hoc networks," in *Proc. Int. Conf. on Sustainable Computing in Science, Technology & Management*, 2019, pp. 1463-1470.

[32] Z. Khan and A. Sharma, "Security aspects of MANETs: A review," *International Journal of Computer Science and Mobile Computing*, vol. 8, no. 7, pp. 40-44, 2019.

[33] F. Abdel-Fattah, K. Farhan, F. Tarawneh, and F. AlTamimi, "Security challenges and attacks in dynamic mobile Ad hoc networks MANETs," in *Proc. IEEE Jordan Int. Joint Conf. on Electrical Engineering and Inform. Techno.*, 2019, pp. 28-33.

[34] B. Alaya and L. Sellami, "Clustering method and symmetric/asymmetric cryptography scheme adapted to securing urban VANET networks," *Journal of Information Security and Applications*, vol. 58, May 2021.

[35] M. Swetha, S. Pushpa, M. Thungamani, T. Manjunath, and S. Deepak, "Strong secure anonymous location based routing (S2ALBR) method for MANET," *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 3, pp. 4349-4356, 2021.

[36] N. Luong, T. Vo, and D. Hoang, "FAPRP: A machine learning approach to flooding attacks prevention routing protocol in mobile ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 2019, pp. 1-17, January 2019.

[37] M. A. Zant and A. Yasin, "Avoiding and isolating flooding attack by enhancing AODV MANET protocol (AIF_AODV)," *Security and Communication Networks*, vol. 2019, pp. 1-12, March 2019.

[38] R. Srilakshmi and M. Bhaskar, "Prevention of attacks in mobile Ad hoc network using african buffalo monitoring zone protocol," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 12, pp. 184-192, 2020.

[39] H. Yang, "A study on improving secure routing performance using trust model in MANET," *Mobile Information Systems*, vol. 2020, pp. 1-17, September 2020.

[40] A. Rajeswari, K. Kulothungan, S. Ganapathy, and A. Kannan, "A trusted fuzzy based stable and secure routing algorithm for effective communication in mobile Ad hoc networks," *Peer-to-Peer Networking and Applications*, vol. 12, pp. 1076-1096, June 2019.

[41] L. Guaya-Delgado, E. Pallarès-Segarra, A. Mezher, and J. Fornés, "A novel dynamic reputation-based source routing protocol for mobile ad hoc networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, pp. 1-16, March 2019.

[42] W. Alnumay, U. Ghosh, and P. Chatterjee, "A trust-based predictive model for mobile Ad hoc network in Internet of things," *Sensors*, vol. 19, no. 6, pp. 1-14, March 2019.

[43] M. Belgaum, S. Musa, M. Su'ud, M. Alam, S. Soomro, and Z. Alansari, "Secured approach towards reactive routing protocols using triple factor in mobile ad hoc networks," *Annals of Emerging Technologies in Computing (AETiC)*, vol. 3, no. 2, pp. 32-40, 2019.

[44] S. Kalime and K. Sagar, "A review: Secure routing protocols for mobile Ad hoc networks (MANETs)," *Journal of Critical Reviews*, vol. 7, no. 19, pp. 8385-8393, 2020.

[45] G. Borkar and A. Mahajan, "A review on propagation of secure data, prevention of attacks and routing in mobile ad-hoc networks," *International Journal of Communication Networks and Distributed Systems*, vol. 24, no. 1, pp. 23-57, 2020.

[46] M. Boulaiche, "Survey of secure routing protocols for wireless Ad hoc networks," *Wireless Personal Communications*, vol. 114, no. 1, pp. 483-517, April 2020.

Liana Khamis Qabajeh received her B.Sc. from Palestine Polytechnic University (PPU), Palestine in 2000 in Computer Engineering and joined the Engineering and Technology Faculty, PPU, as a research assistant. She received her M.Sc. from Jordan University of Science and Technology, Jordan in 2005 in Computer Engineering. Between 2005 and 2008 before pursuing her study, she was primarily involved in academic teaching and research in PPU. She has secured her Ph.D. in Computer Science in 2012 from University of Malaya, Malaysia. In 2012 she has been appointed as assistant professor in Information Technology and Computer Engineering Faculty, PPU. She has been appointed as Master of Informatics program coordinator at PPU during 2016-2019. Her current research interests include Distributed Systems and Ad-Hoc Networks.