ALERT: An Anonymous Location-Based Efficient Routing Protocol in MANETs

Haiying Shen, *Member*, *IEEE*, and Lianyu Zhao, *Student Member*, *IEEE*

Abstract—Mobile Ad Hoc Networks (MANETs) use anonymous routing protocols that hide node identities and/or routes from outside observers in order to provide anonymity protection. However, existing anonymous routing protocols relying on either hop-by-hop encryption or redundant traffic, either generate high cost or cannot provide full anonymity protection to data sources, destinations, and routes. The high cost exacerbates the inherent resource constraint problem in MANETs especially in multimedia wireless applications. To offer high anonymity protection at a low cost, we propose an Anonymous Location-based Efficient Routing proTocol (ALERT). ALERT dynamically partitions the network field into zones and randomly chooses nodes in zones as intermediate relay nodes, which form a nontraceable anonymous route. In addition, it hides the data initiator/receiver among many initiators/receivers to strengthen source and destination anonymity protection. Thus, ALERT offers anonymity protection to sources, destinations, and routes. It also has strategies to effectively counter intersection and timing attacks. We theoretically analyze ALERT in terms of anonymity and efficiency. Experimental results exhibit consistency with the theoretical analysis, and show that ALERT achieves better route anonymity protection and lower cost compared to other anonymous routing protocols. Also, ALERT achieves comparable routing efficiency to the GPSR geographical routing protocol.

Index Terms—Mobile ad hoc networks, anonymity, routing protocol, geographical routing

1 Introduction

Rapid development of Mobile Ad Hoc Networks (MANETs) has stimulated numerous wireless applications that can be used in a wide number of areas such as commerce, emergency services, military, education, and entertainment. MANETs feature self-organizing and independent infrastructures, which make them an ideal choice for uses such as communication and information sharing. Because of the openness and decentralization features of MANETs, it is usually not desirable to constrain the membership of the nodes in the network. Nodes in MANETs are vulnerable to malicious entities that aim to tamper and analyze data and traffic analysis by communication eavesdropping or attacking routing protocols. Although anonymity may not be a requirement in civil-oriented applications, it is critical in military applications (e.g., soldier communication). Consider a MANET deployed in a battlefield. Through traffic analysis, enemies may intercept transmitted packets, track our soldiers (i.e., nodes), attack the commander nodes, and block the data transmission by comprising relay nodes (RN), thus putting us at a tactical disadvantage.

Anonymous routing protocols are crucial in MANETs to provide secure communications by hiding node identities and preventing traffic analysis attacks from outside observers. Anonymity in MANETs includes identity and location anonymity of data sources (i.e., senders) and destinations (i.e., recipients), as well as route anonymity. "Identity and location anonymity of sources and destinations" means it is hard if possible for other nodes to obtain the real identities and exact locations of the sources and destinations. For route anonymity, adversaries, either en route or out of the route, cannot trace a packet flow back to its source or destination, and no node has information about the real identities and locations of intermediate nodes en route. Also, in order to dissociate the relationship between source and destination (i.e., relationship unobservability [1]), it is important to form an anonymous path between the two endpoints and ensure that nodes en route do not know where the endpoints are, especially in MANETs where location devices may be equipped.

Existing anonymity routing protocols in MANETs can be mainly classified into two categories: hop-by-hop encryption [2], [3], [4], [5], [6] and redundant traffic [7], [8], [9], [10], [11], [12], [13]. Most of the current approaches are limited by focusing on enforcing anonymity at a heavy cost to precious resources because public-key-based encryption and high traffic generate significantly high cost. In addition, many approaches cannot provide all of the aforementioned anonymity protections. For example, ALARM [5] cannot protect the location anonymity of source and destination, SDDR [14] cannot provide route anonymity, and ZAP [13] only focuses on destination anonymity. Many anonymity routing algorithms [3], [4], [13], [5], [6], [11], [10] are based on the geographic routing protocol (e.g., Greedy Perimeter Stateless Routing (GPSR) [15]) that greedily forwards a packet to the node closest to the destination. However, the protocol's strict relay node selection makes it easy to reveal the source and destination and to analyze traffic.

On the other hand, limited resource is an inherent problem in MANETs, in which each node labors under an

• *The authors are with the Holcombe Department of Electrical and Computer Engineering, Clemson University, Riggs Hall, Clemson, SC 29634. E-mail: {shenh, lianyuz}@clemson.edu.*

Manuscript received 6 June 2011; revised 24 Nov. 2011; accepted 7 Mar. 2012; published online 13 Mar. 2012.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2011-06-0302. Digital Object Identifier no. 10.1109/TMC.2012.65.

energy constraint. MANETs' complex routing and stringent channel resource constraints impose strict limits on the system capacity. Further, the recent increasing growth of multimedia applications (e.g., video transmission) imposes higher requirement of routing efficiency. However, existing anonymous routing protocols generate a significantly high cost, which exacerbates the resource constraint problem in MANETs. In a MANET employing a high-cost anonymous routing in a battlefield, a low quality of service in voice and video data transmission due to depleted resources may lead to disastrous delay in military operations.

In order to provide high anonymity protection (for sources, destination, and route) with low cost, we propose an Anonymous Location-based and Efficient Routing proTocol (ALERT). ALERT dynamically partitions a network field into zones and randomly chooses nodes in zones as intermediate relay nodes, which form a nontraceable anonymous route. Specifically, in each routing step, a data sender or forwarder partitions the network field in order to separate itself and the destination into two zones. It then randomly chooses a node in the other zone as the next relay node and uses the GPSR [15] algorithm to send the data to the relay node. In the last step, the data is broadcasted to k nodes in the destination zone, providing k-anonymity to the destination. In addition, ALERT has a strategy to hide the data initiator among a number of initiators to strengthen the anonymity protection of the source. ALERT is also resilient to intersection attacks [16] and timing attacks [16]. We theoretically analyzed ALERT in terms of anonymity and efficiency. We also conducted experiments to evaluate the performance of ALERT in comparison with other anonymity and geographic routing protocols. In summary, the contribution of this work includes:

1. *Anonymous routing.* ALERT provides route anonymity, identity, and location anonymity of source and destination.
2. *Low cost.* Rather than relying on hop-by-hop encryption and redundant traffic, ALERT mainly uses randomized routing of one message copy to provide anonymity protection.
3. *Resilience to intersection attacks and timing attacks.* ALERT has a strategy to effectively counter intersection attacks, which have proved to be a tough open issue [16]. ALERT can also avoid timing attacks because of its nonfixed routing paths for a source-destination pair.
4. *Extensive simulations.* We conducted comprehensive experiments to evaluate ALERT's performance in comparison with other anonymous protocols.

The remainder of this paper is organized as follows: In Section 2, we present the design of the ALERT routing protocol. Section 3 discusses the anonymity performance of ALERT and its strategies to deal with certain attacks. In Section 4, we theoretically analyzed ALERT in terms of anonymity and efficiency. Experimental performance of the ALERT protocol is evaluated in Section 5. In Section 6, we describe related anonymous routing approaches in MANETs. The conclusion and future work are given in Section 7.

2 ALERT: An Anonymous Location-Based Efficient Routing Protocol

2.1 Networks and Attack Models and Assumptions

ALERT can be applied to different network models with various node movement patterns such as random way point model [17] and group mobility model [18]. Consider a MANET deployed in a large field where geographic routing is used for node communication in order to reduce the communication latency. The location of a message's sender may be revealed by merely exposing the transmission direction. Therefore, an anonymous communication protocol that can provide untraceability is needed to strictly ensure the anonymity of the sender when the sender communicates with the other side of the field. Moreover, a malicious observer may try to block the data packets by compromising a number of nodes, intercept the packets on a number of nodes, or even trace back to the sender by detecting the data transmission direction. Therefore, the route should also be undetectable. A malicious observer may also try to detect destination nodes through traffic analysis by launching an intersection attack. Therefore, the destination node also needs the protection of anonymity.

In this work, the attackers can be battery powered nodes that passively receive network packets and detect activities in their vicinity. They can also be powerful nodes that pretend to be legitimate nodes and inject packets to the network according to the analytical results from their eavesdropped packets. The assumptions below apply to both inside and outside attackers.

1. *Capabilities.* By eavesdropping, the adversary nodes can analyze any routing protocol and obtain information about the communication packets in their vicinity and positions of other nodes in the network. They can also monitor data transmission on the fly when a node is communicating with other nodes and record the historical communication of nodes. They can intrude on some specific vulnerable nodes to control their behavior, e.g., with denial-of-service (DoS) attacks, which may cut the routing in existing anonymous geographic routing methods.
2. *Incapabilities.* The attackers do not issue strong active attacks such as black hole. They can only perform intrusion to a proportion of all nodes. Their computing resources are not unlimited; thus, both symmetric and public/private key cannot be brutally decrypted within a reasonable time period. Therefore, encrypted data are secure to a certain degree when the key is not known to the attackers.

2.2 Dynamic Pseudonym and Location Service

In one interaction of node communication, a source node S sends a request to a destination node D and the destination responds with data. A *transmission session* is the time period that S and D interact with each other continuously until they stop. In ALERT, each node uses a dynamic pseudonym as its node identifier rather than using its real MAC address, which can be used to trace nodes' existence in the network. To avoid pseudonym collision, we use a collision-resistant hash function, such as SHA-1 [19], to hash a node's

MAC address and current time stamp. To prevent an attacker from recomputing the pseudonym, the time stamp should be precise enough (e.g., nanoseconds). Considering the network delay, the attacker needs to compute, e.g., 10^5, times for one packet per node. There may also be many nodes for an attacker to listen, so the computing overhead is not acceptable, and the success rate is low. To further make it more difficult for an attacker to compute the times tamp, we can increase the computation complexity by using randomization for the time stamps. Specifically, we keep the precision of time stamp to a certain extent, say 1 second, and randomize the digits within 1/10th. Thus, the pseudonyms cannot be easily reproduced. A node's pseudonym expires after a specific time period in order to prevent adversaries from associating the pseudonyms with nodes. If pseudonyms are changed too frequently, the routing may get perturbed; and if pseudonyms are changed too infrequently, the adversaries may associate pseudonyms with nodes across pseudonym changes. Therefore, the pseudonym change frequently should be appropriately determined. Each node periodically piggybacks its updated position and pseudonym to "hello" messages, and sends the messages to its neighbors. Also, every node maintains a routing table that keeps its neighbors' pseudonyms associated with their locations.

As previous works [10], [13], we assume that the public key and location of the destination of a data transmission can be known by others, but its real identity requires protection. We can utilize a secure location service [20], [21], [22], [23], [24] to provide the information of each node's location and public key. Such a location service enables a source node, who is aware of the identity of the destination node, to securely obtain the location and public key of the destination node. The public key is used to enable two nodes to securely establish a symmetric key K_s for secure communication. The destination location enables a node to determine the next hop in geographic routing. Specifically, trusted normal nodes or dedicated service provider nodes are used to provide location service. Each node has a location server. When a node A wants to know the location and public key of another node B, it will sign the request containing B's identity using its own identity. Then, the location server of A will return an encrypted position of B and its public key, which can be decrypted by A using the predistributed shared key between A and its location server. When node A moves, it will also periodically update its position to its location server.

For high reliability, the location serves can replicate data between each other. Thus, the location servers are allowed to fail, because each node can be in contact with all location servers in range. For example, current location service solutions such as [20] are able to seamlessly let node switch between location servers. We assume that the attacker will not compromise and utilize the location to find out the real identities of nodes that contact with the compromised location server, which is the common assumption of current location services [20], [21]. We leave the work on secure location service as our future.

The existence of the location servers are opposed to the ad hoc property of MANETs, and it is not necessary to use location servers in a MANET without security consideration. However, anonymous communication requires third-party servers to reliably collect and transmit confidential information, and this solution was used in many of previously proposed works such as [4], [11], [13]. With the advance of wireless access point (AP), the deployment of location services can be conducted by placing several APs in the whole WIMAX network of civil use at a reasonable cost. It is difficult to preserve all stable location servers in a battle field, but since the location servers are not necessarily be functional all the time and each node only needs to have one usable location server, the location servers can be buried under the ground where anonymous communication is needed.

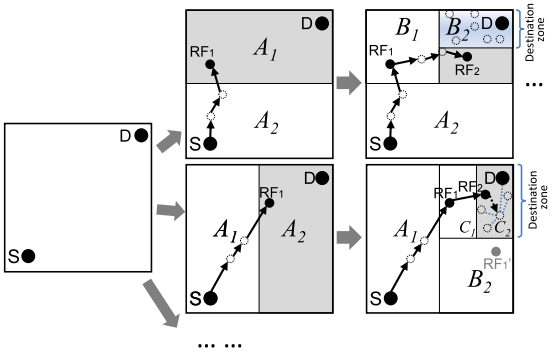


Fig. 1. Examples of different zone partitions.

2.3 The ALERT Routing Algorithm

For ease of illustration, we assume the entire network area is generally a rectangle in which nodes are randomly disseminated. The information of the bottom-right and upper left boundary of the network area is configured into each node when it joins in the system. This information enables a node to locate the positions of nodes in the entire area for zone partitions in ALERT.

ALERT features a dynamic and unpredictable routing path, which consists of a number of dynamically determined intermediate relay nodes. As shown in the upper part of Fig. 1, given an area, we horizontally partition it into two zones A_1 and A_2. We then vertically partition zone A_1 to B_1 and B_2. After that, we horizontally partition zone B_2 into two zones. Such zone partitioning consecutively splits the smallest zone in an alternating horizontal and vertical manner. We call this partition process *hierarchical zone partition*. ALERT uses the hierarchical zone partition and randomly chooses a node in the partitioned zone in each step as an intermediate relay node (i.e., data forwarder), thus dynamically generating an unpredictable routing path for a message.

Fig. 2 shows an example of routing in ALERT. We call the zone having k nodes where D resides the *destination zone*, denoted as Z_D. k is used to control the degree of anonymity protection for the destination. The shaded zone in Fig. 2 is the destination zone. Specifically, in the ALERT routing, each data source or forwarder executes the hierarchical zone partition. It first checks whether itself and destination are in the same zone. If so, it divides the zone alternatively in the horizonal and vertical directions. The node repeats this

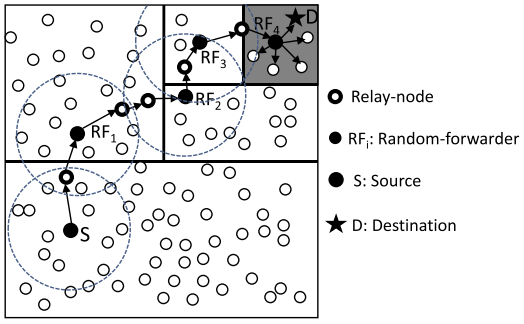


Fig. 2. Routing among zones in ALERT.

process until itself and Z_D are not in the same zone. It then randomly chooses a position in the other zone called *temporary destination* (TD), and uses the GPSR routing algorithm to send the data to the node closest to *TD*. This node is defined as a random forwarder (*RF*). Fig. 3 shows an example where node N_3 is the closest to *TD*, so it is selected as a RF. ALERT aims at achieving k-anonymity [25] for destination node D, where k is a predefined integer. Thus, in the last step, the data are broadcasted to k nodes in Z_D, providing k-anonymity to the destination.

Given an S-D pair, the partition pattern in ALERT varies depending on the randomly selected *TD*s and the order of horizontal and vertical division, which provides a better anonymity protection. Fig. 1 shows two possible routing paths for a packet pkt issued by sender S targeting destination D in ALERT. There are also many other possible paths. In the upper routing flow, data source S first horizontally divides the area into two equal-size zones, A_1 and A_2, in order to separate S and Z_D. S then randomly selects the first temporary destination TD_1 in zone A_1 where Z_D resides. Then, S relies on GPSR to send pkt to TD_1. The pkt is forwarded by several relays until reaching a node that cannot find a neighbor closer to TD_1. This node is considered to be the first random-forwarder RF_1. After RF_1 receives pkt, it vertically divides the region A_1 into regions B_1 and B_2 so that Z_D and itself are separated in two different zones. Then, RF_1 randomly selects the next temporary destination TD_2 and uses GPSR to send pkt to TD_2. This process is repeated until a packet receiver finds itself residing in Z_D, i.e., a partitioned zone is Z_D having k nodes. Then, the node broadcasts the pkt to the k nodes.

The lower part of Fig. 1 shows another routing path based on a different partition pattern. After S vertically partitions the whole area to separate itself from Z_D, it randomly chooses TD_1 and sends pkt to RF_1. RF_1 partitions zone A_2 into B_1 and B_2 horizontally and then partitions B_1 into C_1 and C_2 vertically, so that itself and Z_D are separated. Note that RF_1 could vertically partition A_2 to separate itself from Z_D in two zones but may choose a TD further away from the destination than the TD that resulted from the horizontal partition. Therefore, ALERT sets the partition in the alternative horizontal and vertical manner in order to ensure that a pkt approaches D in each step.

As GPSR, we assume that the destination node will not move far away from its position during the data transmission, so it can successfully receive the data. In this design, the tradeoff is the anonymity protection degree and

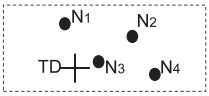


Fig. 3. Choosing a *RF* according to a given *TD*.

transmission delay. A larger number of hierarchies generate more routing hops, which increases anonymity degree but also increases the delay. To ensure the delivery of packets, the destination sends a confirmation to the source upon receiving the packets. If the source has not received the confirmation during a predefined time period, it will resend the packets.

2.4 The Destination Zone Position

The reason we use Z_D rather than D is to avoid exposure of D. *Zone position* refers to the upper left and bottom-right coordinates of a zone. One problem is how to find the position of Z_D, which is needed by each packet forwarder to check whether it is separated from the destination after a partition and whether it resides in Z_D. Let H denote the total number of partitions in order to produce Z_D. Using the number of nodes in Z_D (i.e., k), and node density ρ, H is calculated by

$$H = log_2\left(\frac{\rho \cdot G}{k}\right),$$

where G is the size of the entire network area. Using the calculated H, the size G, the positions $(0,0)$ and (x_G, y_G) of the entire network area, and the position of D, the source S can calculate the zone position of Z_D. Assume ALERT partitions zone vertically first. After the first vertical partition, the positions of the two generated zones are $(0,0),(0.5x_G, y_G)$ and $(0.5x_G, 0),(x_G, y_G)$. S then finds the zone where Z_D is located and divides that zone horizontally. This recursive process continues until H partitions are completed. The final generated zone is the desired destination zone, and its position can be retrieved accordingly. Therefore, the size of the destination zone is $\frac{G}{2^H}$. For example, for a network with size $G = 8$ and position represented by $(0,0)$ and $(4,2)$, if $H = 3$ and the destination position is $(0.5, 0.8)$, the resulting destination zone's position is $(0,0)$ and $(1,1)$ with size of $\frac{8}{2^3} = 1$.

2.5 Packet Format of ALERT

For successful communication between S and D, S and each packet forwarder embeds the following information into the transmitted packet.

1. The zone position of Z_D, i.e., the Hth partitioned zone.
2. The encrypted zone position of the Hth partitioned zone of S using D's public key, which is the destination for data response.
3. The current randomly selected TD for routing.
4. A bit (i.e., 0/1), which is flipped by each *RF*, indicating the partition direction (horizontal or vertical) of the next *RF*.

With the encrypted Hth partitioned zone in the information of (2), an attacker needs very high computation power to be able to launch attacks such as dictionary attack to

RREQ/RREP/NAK		P_S	P_D	L_{z_S}	L_{z_D}	L_{RF}
h	H	K^S_{pub}	$(TTL)_{K^{RN}_{pub}}$	$(Bitmap)_{K^D_{pub}}$		data (NULL in NAK)

Fig. 4. Packet format of ALERT.

decrypt it in order to discover the source S of a session with a specific destination D. Moreover, the Hth partitioned zone is the position of a zone rather than a position, which makes it even harder to locate the source S. Such an attack from an attacker with very high computation power is beyond our practical assumption.

In order to save computing resources, we let the source node calculate the information of (1) and (2) and forward it along the route rather than letting each packet forwarder calculate the values. In order to hide the packet content from adversaries, ALERT employs cryptography. The work in [26] experimentally proved that generally symmetric key cryptography costs hundreds of times less overhead than public key cryptography while achieving the same degree of security protection. Thus, instead of using public key cryptography, ALERT uses symmetric key encryption for transmitted data. Recall that S can get D's public key from the secure location service. In a S-D communication, S first embeds a symmetric key K^S_s, encrypted using D's public key, into a packet. Later, D sends S its requested contents, encrypted with K^S_s, decrypted by its own public key. Therefore, the packets communicated between S and D can be efficiently and securely protected using K^S_s.

Fig. 4 shows the packet format of ALERT, which omits the MAC header. Because of the randomized routing nature in ALERT, we have a universal format for RREQ/RREP/NAK. A node use NAK to acknowledge the loss of packets. The data field of RREQ/RREP is left blank in NAK packets. Flooding-based anonymity routing usually uses ACKs, while NAKs are often adopted in geographic routing-based approaches [13] to reduce traffic cost. For the same purpose, we choose to use NAKs. In the packet, P_S is the pseudonym of a source; P_D is the pseudonym of the destination; L_{Z_S} and L_{Z_D} are the positions of the Hth partitioned source zone and destination zone, respectively; L_{TD} is the currently selected TD's coordinate; h is the number of divisions so far, H is the maximum allowed number of divisions; and K^S_s denotes the symmetric key of a source. Particularly, $(TTL)_{K^{RN}_{pub}}$ is used for the protection of source anonymity and will be introduced in Section 2.6, and $(Bitmap)_{K^D_{pub}}$ is used for solving intersection attack and will be discussed in Section 3.3. When node A wants to know the location and public key of another node B, it will contact its location server as described in Section 2.2, thus there is no need to exchange shared keys between nodes.

2.6 Source Anonymity

ALERT contributes to the achievement of anonymity by restricting a node's view only to its neighbors and constructing the same initial and forwarded messages. This makes it difficult for an intruder to tell if a node is a source or a forwarding node. To strengthen the anonymity protection of the source nodes, we further propose a lightweight mechanism called "notify and go." Its basic idea is to let a number of nodes send out packets at the same time as S in order to hide the source packet among many other packets.

"Notify and go" has two phases: "notify" and "go." In the first "notify" phase, S piggybacks its data transmission notification with periodical update packets to notify its neighbors that it will send out a packet. The packet includes two random back-off time periods, t and t_0. In the "go" phase, S and its neighbors wait for a certain period of randomly chosen time $\in [t, t+t_0]$ before sending out messages. S's neighbors generate only several bytes of random data just in order to cover the traffic of the source. t should be a small value that does not affect the transmission latency. A long t_0 may lead to a long transmission delay while a short t_0 may result in interference due to many packets being sent out simultaneously. Thus, t_0 should be long enough to minimize interference and balance out the delay between S and S's farthest neighbor in order to prevent any intruder from discriminating S. This camouflage augments the privacy protection for S by η-anonymity where η is the number of its neighbors. Therefore, it is difficult for an attacker to analyze traffic to discover S even if it receives the first notification.

ALERT utilizes a TTL field in each packet to prevent the packets issued in the first phase from being forwarded in order to reduce excessive traffic. Only the packets of S are assigned a valid TTL, while the covering packets only have a $\mathrm{TTL} = 0$. After S decides the next TD, it forwards the packet to the next relay node, which is its neighbor based on GPSR. To prevent the covering packets from being differentiated from the ones sent by S, S encrypts the TTL field using K^{RN}_{pub} obtained from the periodical "hello" packets between neighbors. Every node that receives a packet but cannot find a valid TTL will try to decrypt the TTL using its own private key. Therefore, only NRN will be able to successfully decrypt it, while other nodes will drop such a packet.

2.7 Will Dead End Compromise Anonymity?

Dead end is one common problem in the geographic routing in which each node is aware of the positions of its neighbors in order to forward a packet to the neighbor nearest to the destination. A dead end occurs when a packet is forwarded to a node whose neighbors are all further away from the destination than itself and then the packet is routed between neighbors iteratively. ALERT can incorporate existing solutions [24], [27], [28], such as face routing, to avoid the dead-end problem without compromising anonymity protection. In ALERT, the transmission of each packet is based on a series of RFs who decide which region a packet should be sent to. Between any two RFs, the relays perform the GPSR routing. Each relay has no information on the S or D except the destination zone information. Its routing action is based on the coordinate of the next TD. Therefore, relays can incorporate existing solutions to avoid the dead-end problem without exposing any direct information about the S or D.

3 Anonymity Protection and Strategies against Attacks

This section discusses the performance of ALERT in providing anonymity protection and its performance and strategies to deal with some attacks.

3.1 Anonymity Protection

ALERT offers identity and location anonymity of the source and destination, as well as route anonymity. Unlike geographic routing [29], [3], [4], [10], [11], which always takes the shortest path, ALERT makes the route between a S-D pair difficult to discover by randomly and dynamically selecting the relay nodes. The resultant different routes for transmissions between a given S-D pair make it difficult for an intruder to observe a statistical pattern of transmission. This is because the *RF* set changes due to the random selection of *RF*s during the transmission of each packet. Even if an adversary detects all the nodes along a route once, this detection does not help it in finding the routes for subsequent transmissions between the same S-D pair.

Additionally, since an *RF* is only aware of its proceeding node and succeeding node in route, the source and destination nodes cannot be differentiated from other nodes en route. Also, the anonymous path between S and D ensures that nodes on the path do not know where the endpoints are. ALERT strengthens the privacy protection for S and D by the *unlinkability* of the transmission endpoints and the transmitted data [1]. That is, S and D cannot be associated with the packets in their communication by adversaries. ALERT incorporates the "notify and go" mechanism to prevent an intruder from identifying which node within the source neighborhood has initiated packets. ALERT also provides k-anonymity to destinations by hiding D among k receivers in Z_D. Thus, an eavesdropper can only obtain information on Z_D, rather than the destination position, from the packets and nodes en route.

The route anonymity due to random relay node selection in ALERT prevents an intruder from intercepting packets or compromising vulnerable nodes en route to issue DoS attacks. In ALERT, the routes between two communicating nodes are constantly changing, so it is difficult for adversaries to predict the route of the next packet for packet interception. Similarly, the communication of two nodes in ALERT cannot be completely stopped by compromising certain nodes because the number of possible participating nodes in each packet transmission is very large due to the dynamic route changes. In contrast, these attacks are easy to perform in geographic routing, since the route between a given S-D pair is unlikely to change for different packet transmissions, and thus, the number of involved nodes is much smaller than in ALERT.

3.2 Resilience to Timing Attacks

In timing attacks [16], through packet departure and arrival times, an intruder can identify the packets transmitted between S and D, from which it can finally detect S and D. For example, two nodes A and B communicate with each other at an interval of 5 seconds. After a long observation time, the intruder finds that A's packet sending time and B's packet receiving time have a fixed five second difference such as (19:00:55, 19:01:00) and (20:01:33, 20:01:38). Then, the intruder would suspect that A and B are communicating with each other.

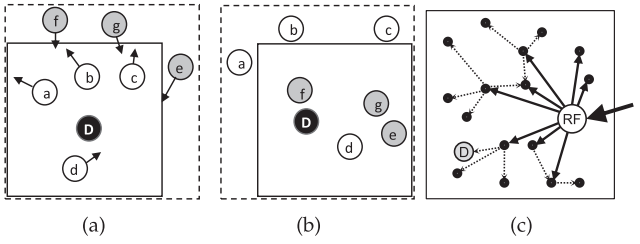


Fig. 5. Intersection attack and solution.

Avoiding the exhibition of interaction between communication nodes is a way to counter timing attacks. In ALERT, the "notify and go" mechanism and the broadcasting in Z_D both put the interaction between S-D into two sets of nodes to obfuscate intruders. More importantly, the routing path between a given S-D and the communication delay (i.e., time stamp) change constantly, which again keeps an intruder from identifying the S and D.

3.3 Strategy to Counter Intersection Attacks

In an intersection attack, an attacker with information about active users at a given time can determine the sources and destinations that communicate with each other through repeated observations. Intersection attacks are a well-known problem and have not been well resolved [16]. Though ALERT offers k-anonymity to D, an intersection attacker can still identify D from repeated observations of node movement and communication if D always stays in Z_D during a transmission session. This is because as long as D is conducting communication, the attacker can monitor the change of the members in the destination zone containing D. As time elapses and nodes move, all other members may move out of the destination zone except D. As a result, D is identified as the destination because it always appears in the destination zone.

Fig. 5a is the status of a Z_D after a packet is broadcasted to the zone. The arrows show the moving directions of nodes. We can see that nodes a, b, c, d, and D are in Z_D. Fig. 5b is the subsequent status of the zone the next time a packet is transmitted between the same S-D pair. This time, nodes d, e, f, g, and D are in Z_D. Since the intersection of the in-zone nodes in both figures includes d and D, D could be identified by the attacker. Therefore, the longer an attacker watches the process, the easier it is to identify the destination node.

To counter the intersection attack, ZAP [13] dynamically enlarges the range of anonymous zones to broadcast the messages or minimizes communication session time. However, the former strategy increases the communication overhead, while the latter may not be suitable for long-duration communication. Instead of adopting such a mitigating mechanism, we propose another strategy to resolve this problem. Note that the attacker can be puzzled and lose the cumulated observation by making it occasionally *fail* to observe D's reception of packets. Since packets are delivered to Z_D constantly in long-duration sessions rather than using direct local broadcasting in the zone, the last *RF* multicasts packet pkt_1 to a partial set of nodes, say

m nodes out of the total k nodes in the zone. The m nodes hold the packets until the arrival of the next packet pkt_2. Upon the arrival of the next packet, the m nodes conduct one-hop broadcasting to enable other nodes in the zone to also receive the packet in order to hide D.

Fig. 5c shows the two-step process with the first step in solid arrows and the second step in dashed arrows. We can see that the first step reaches a number of nodes in the destination zone, but the destination is reached in the second step. Because the deliveries of pkt_1 and pkt_2 are mixed, an attacker observes that D is not in the recipient set of pkt_1 though D receives pkt_1 in the delivery time of pkt_2. Therefore, the attacker would think that D is not the recipient of every packet in Z_D in the transmission session, thus foiling the intersection attack.

Because the attacker may grab and analyze packets on air, the last forwarding node alters a number of bits in each packet to prevent the attacker from identifying identical packets in one broadcasting. This function is provided by the field $(Bitmap)_{K_{pub}^D}$ in each packet. The $Bitmap$ records the altered bits and is encrypted using the destination's public key K_{pub}^D for recovering the original data. Since destination is not always within the recipient set, and the packet forwarded to a destination is different from the original packet, the attacker cannot identify the destination from its observation history by calculating the intersection set of nodes. This approach incurs two extra costs. One is the one-hop broadcasting of the recipients in the destination zone. The other is the encryption cost of changed bits.

The percentage of nodes in Z_D that can receive the packet (i.e., coverage percent) is $\frac{m}{k} + (1 - \frac{m}{k}) \times p_c = p_c + m \times \frac{1-p_c}{k}$, where p_c denotes the percentage of the $k - m$ nodes that receive the packet from the m nodes in the second step. To ensure that D receives the packet, p_c should equal 1. $p_c = 1$ can be achieved by a moderate value of m considering node transmission range. A lower transmission range leads to a higher value of m and vice versa.

4 Theoretical Analysis

In this section, we theoretically analyze the anonymity and routing efficiency properties of ALERT. We analyze the number of nodes that can participate in routing that function as camouflages for routing nodes. We estimate the number of *RF*s in a routing path, which shows the route anonymity degree and routing efficiency of ALERT. We calculate the anonymity protection degree of a destination zone as time passes to demonstrate ALERT's ability to counter intersection attacks. In this section, we also use figures to show the analytical results to clearly demonstrate the relationship between these factors and the anonymity protection degree.

In our analysis scenario, we assume that the entire network area is a rectangle with side lengths l_A and l_B and the entire area is partitioned H times to produce a k-anonymity destination zone. For the parameters of results in the figures, unless otherwise indicated, the size of the entire network zone is $1{,}000\ \mathrm{m} \times 1{,}000\ \mathrm{m}$ and the number of nodes equals 200. We set $H = 5$ to ensure that a reasonable number of nodes are in a destination zone.

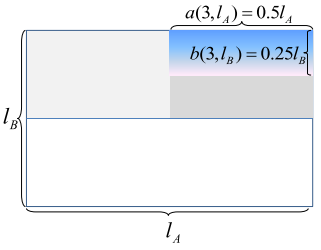


Fig. 6. The side lengths of the 3rd partitioned zone.

We first introduce two functions to calculate the two side lengths of the hth partitioned zone:

$$a(h, l_A) = \frac{l_A}{2^{\lfloor h/2 \rfloor}}, \tag{1}$$

$$b(h, l_B) = \frac{l_B}{2^{\lceil h/2 \rceil}}. \tag{2}$$

The side lengths of the destination zone after H partitions are $a(H, l_A)$ and $b(H, l_B)$. Fig. 6 shows an example of three partitions of the entire network area. The side lengths of the final zone after the three partitions are

$$a(3, l_A) = \frac{l_A}{2^{\lfloor 3/2 \rfloor}} = 0.5 l_A \tag{3}$$

and

$$b(3, l_B) = \frac{l_B}{2^{\lceil 3/2 \rceil}} = 0.25 l_B. \tag{4}$$

4.1 The Number of Possible Participating Nodes

The intention of this analysis is to characterize how many possible nodes are able to participate in one S-D routing. The number of these nodes shows how many nodes can become camouflages in a routing path. These possible participating nodes include *RF*s and the relay nodes between two *RF*s using GPSR. The nodes that actually conduct the routing are not easily discovered among the many possible participating nodes, thus making the routing pattern undetectable. Because the positions of both S and D affect the number of possible participating nodes in routing, the positions influence routing anonymity.

We first calculate the probability that σ partitions are needed to separate S and D denoted as $p_s(\sigma)$. We use σ to denote the *closeness* between S and D. $p_s(\sigma)$ actually is the probability that D is located in a position that can be separated from a given S using σ partitions. We can get

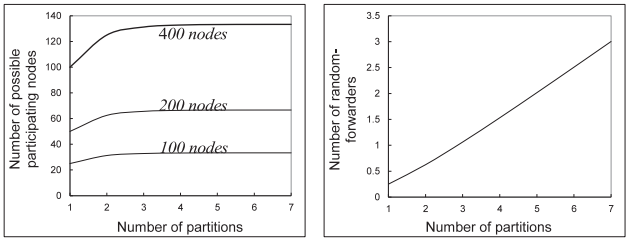
$$p_s(\sigma) = \frac{1}{2^\sigma}, \quad 0 < \sigma \leq H. \tag{5}$$

We use $N_e(\sigma)$ to denote the expected number of nodes that possibly take part in routing based on a given closeness σ:

$$N_e(\sigma) = a(\sigma, l_A) b(\sigma, l_B) \rho, \tag{6}$$

where ρ denotes the density of nodes. By considering different closeness σ, we arrive at the final expected number of possible participating nodes from a S to any D:

$$N_e = \sum_{\sigma=1}^{H} N_e(\sigma) p_s(\sigma) = \sum_{\sigma=1}^{H} (a(\sigma, l_A) b(\sigma, l_B) \rho) \frac{1}{2^\sigma}. \tag{7}$$



(a) Estimated possible participat- (b) Estimated random-forwarders.

Fig. 7. Estimated routing nodes.

We set the total number of nodes in the network to 100, 200, and 400, respectively, and use (5) to calculate the number of possible participating nodes. Fig. 7a shows the result versus the number of partitions. We observe that the number of possible partitioning nodes exhibits a relatively faster increase when the number of partitions H increases from 1 to 2. Later on, as H increases, the increase speed of the number slows down and tends to maintain around a constant (about 1/4 of the total number of nodes). When $H = 1$, the destination zone is large, and the probability that D is located in a position that can be separated from a given S using one partition, $p_s(\sigma)$, equals $\frac{1}{2}$. The probability that D and S cannot be separated using one partition is also $\frac{1}{2}$, and in this case, no random forwarders are needed, leading to a relatively low number of possible participating nodes. When H increases to 2, S and D have higher probability to be separated. Thus, more random forwarders are selected and the number of possible participating nodes increases. As H continues to increase, the probability that S and D need H partitions to separate from each other ($p_s(\sigma)$) exponentially decreases, and then the number of possible participating nodes increases more and more slowly. The result implies that H should be appropriately determined in order to balance the tradeoff between the degree of anonymity protection and routing cost.

4.2 The Number of Random-Forwarders

The number of *RF*s determines the length of the routing path in ALERT. Therefore, it reflects the energy efficiency and degree of anonymity of ALERT. From the anonymity view, for a network with a fixed number of nodes, more *RF*s offer higher anonymity but will reduce the number of nodes in the destination zone, and consequently reduce the anonymity protection of destination node. Therefore, the number of *RF*s should be carefully determined to ensure a sufficient number of nodes located in the destination zone. For a pair of S-D with closeness σ, we define $p_i(\sigma, i)$ as the probability that an S-D routing path has i *RF*s. The number of *RF*s is determined by the zone partition pattern. For example, in the bottom-right part in Fig. 1, the random forwarder randomly chosen by S could be in area B_2 or C_1. If S chooses $RF_1{}'$ in area B_2, a random forwarder in area C_1 may be subsequently chosen. If S chooses RF_1 in area C_1, it loses the opportunity to select an RF in area B_2, which means this routing has one RF less than the previous case. We use RF^+ to represent the former case and use RF^- to represent the latter case.

Recall that H is the maximum number of partitions for an S-D pair. σ is the number of partitions performed before the first *RF* is chosen. Therefore, $p_i(\sigma, i)$ is determined by i *RF* choices that lead to RF^+, and by $(H - \sigma - i)$ *RF* choices that lead to RF^-. For each *RF* selection, it has $\frac{1}{2}$ probability to result in RF^-, and also has $\frac{1}{2}$ probability to result in RF^+. Moreover, $p_i(\sigma, i)$ is only related to the number of RF^+s and RF^-s instead of the sequence of such RF^+ and RF^- choices, so it fits the Binomial distribution.

Therefore,

$$p_i(\sigma, i) = C^i_{H-\sigma}\left(\frac{1}{2}\right)^i\left(\frac{1}{2}\right)^{H-\sigma-i} = C^i_{H-\sigma}\left(\frac{1}{2}\right)^{H-\sigma}. \quad (8)$$

Using $N_{RF}(\sigma)$ to denote the expected number of *RF*s, we get

$$N_{RF}(\sigma) = \sum_{i=1}^{H-\sigma} p_i(\sigma, i)i = \sum_{i=1}^{H-\sigma} C^i_{H-\sigma}\left(\frac{1}{2}\right)^{H-\sigma} i. \quad (9)$$

Finally, considering the probability of different closeness between S and D in (5), we have

$$N_{RF} = \sum_{\sigma=1}^{H}\sum_{i=1}^{H-\sigma} C^i_{H-\sigma}\left(\frac{1}{2}\right)^{H-\sigma}\frac{i}{2^\sigma}, \quad (10)$$

where N_{RF} denotes the expected number of *RF*s.

Fig. 7b demonstrates the number of *RF*s versus the number of partitions calculated using (8). The result indicates that the number of *RF*s increases linearly as the number of partitions increases. This is because one partition may generate one more *RF*, and more partitions generate more *RF*s. For a S-D pair, each partition leads to either RF^+ or RF^- with the same probability. Since the final number of *RF*s depends on RF^+ choices, the number of *RF*s increases proportionally as H increases. The figure shows the number of *RF*s by considering different location relationships of S and D.

More *RF*s in a routing path provide higher anonymity protection to the route and two endpoints. Although we can achieve a large number of *RF*s with a large H, the zone for *RF* selection becomes smaller and smaller. Thus, the number of options for random *RF* selection decreases, which means that the anonymity protection is enhanced with a decreasing speed. This means it is important to decide an appropriate H that can achieve high anonymity protection without incurring significant overhead due to many partitions and long path length.

4.3 Destination Anonymity Protection

Destination anonymity is determined by the number of nodes in the destination zone, which is related to node density and the size of the destination zone. According to the work in [13], the probability that a node with a moving speed v remains in the destination zone, which is a circular area with radius r, after time period t, denoted by $p_r(t)$, is exponentially distributed:

$$p_r(t) = e^{-t/\beta(r)}, \quad (11)$$

where

$$\beta(r) = \frac{\pi r}{2v}. \quad (12)$$

In order to apply (11) and (12) to our method, we assume the Hth partitioned destination zone is a square that can be approximated by a circle covering approximately (see Fig. 8)

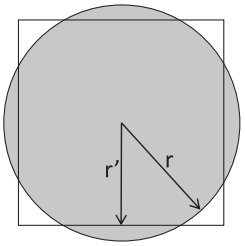
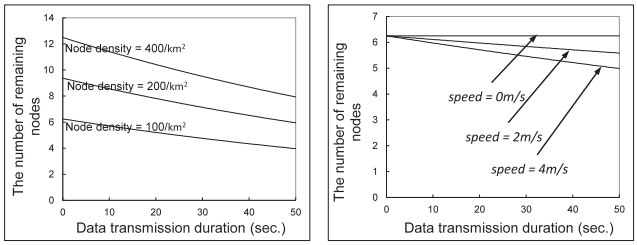


Fig. 8. Approximating a zone using a circle.



(a) Density variation (b) Speed variation

Fig. 9. Estimated destination anonymity protection.

the same area. This assumption is feasible, which only requires a square for the entire network area (i.e., $l_A = l_B$) and an even number of partitions (i.e., $a(H, l_A) = b(H, l_A)$). We use $2r'$ to denote the side length of the destination zone. Hence, we can calculate the radius of this approximate circle as below:

$$\pi r^2 = (2r')^2 \rightarrow r = \frac{2r'}{\sqrt{\pi}}. \tag{13}$$

Thus,

$$\beta(r) = \frac{\sqrt{\pi} r'}{v}. \tag{14}$$

We use $N_r(t)$ to denote the number of nodes remaining in the destination zone after a time period t. Then, we have

$$N_r(t) = p_r(t) a(H, l_A) b(H, l_B) \rho = e^{\frac{-tv}{\sqrt{\pi} r'}} a(H, l_A)^2 \rho. \tag{15}$$

According to (11) and (14), $p_r(t)$ is only related to the nodes' moving speed v and the destination zone side length $2r'$. Then, we can conclude that for a network with a given node density, in ALERT, the number of remaining nodes in the destination zone decreases constantly as time goes on. After time t, it is determined only by nodes' moving speed and the size of the destination zone.

We use (13) to calculate the number of remaining nodes in a destination zone (*remaining nodes* for short) based on node density and moving speed. We assume the process of node movement in a destination zone follows exponential distribution [13]. We use the term *data transmission duration* to denote the elapsed time of node communicates. Fig. 9a shows the number of remaining nodes at node moving speed of 2 m/s with node densities 100, 200, and 400 over time. As node density decreases, the number of remaining nodes drops and hence the degree of destination anonymity protection decreases correspondingly. Also, the number of remanet nodes decreases as the time increases. Fig. 9b demonstrates the number of remaining nodes when the node density equals to $200/\mathrm{km}^2$ versus different node moving speed over time. The number of remaining nodes decreases when the speed of nodes in the destination zone becomes faster, so the degree of anonymity protection of the destination decreases. Moreover, the number drops as time elapses due to the node mobility. The figures confirm our conclusion that the anonymity protection of the destination is determined by the nodes' moving speed and the size of the destination zone, and the protection degree decreases constantly over time.

For ALERT to be usable, we need to ensure that the pseudonym and location exchange cost is low compared with regular communication messages. Let N, N_L, f, F, and T denote the total number of nodes, the number of location servers, the frequency of pseudonym, and location updates and the frequency of regular communication messages, respectively. The number of messages exchanged between location servers within time T is $N_L \times (N_L - 1) \times f \times T$, the number of messages for pseudonym updates is $N \times f \times T$. The number of communication messages in the network is $N \times F \times T$. Therefore, if the location servers incur only a small fraction of messages, we need to make sure that $\frac{N_L \times (N_L - 1) \times f \times T + N \times f \times T}{N \times F \times T} \ll 1$. Regular communication frequency should be much higher than update exchange messages. Thus, $f \ll F$, so that $\frac{N \times f \times T}{N \times F \times T} \ll 1$. Therefore,

$$\begin{aligned} &\frac{N_L \times (N_L - 1) \times f \times T + N \times f \times T}{N \times F \times T} \ll 1 \\ &\rightarrow \frac{N_L \times (N_L - 1) \times f \times T}{N \times F \times T} \ll 1 \\ &\rightarrow \frac{N_L \times (N_L - 1) \times f}{N \times F} \ll 1, \end{aligned}$$

which can be satisfied if N_L is comparable to $\sqrt{N}$. This is reasonable when the transmission range of nodes is modest so that only a small number of location servers are needed.

5 Performance Evaluation

In this section, we provide experimental evaluation of the ALERT protocol, which exhibit consistency with our analytical results. Both prove the superior performance of ALERT in providing anonymity with low cost of overhead. Recall that anonymous routing protocols can be classified into hop-by-hop encryption and redundant traffic. We compare ALERT with two recently proposed anonymous geographic routing protocols: AO2P [10] and ALARM [5], which are based on hop-by-hop encryption and redundant traffic, respectively. All of the protocols are geographic routing, so we also compare ALERT with the baseline routing protocol GPSR [30] in the experiments. In GPSR, a packet is always forwarded to the node nearest to the destination. When such a node does not exist, GPSR uses perimeter forwarding to find the hop that is the closest to the destination. In ALARM, each node periodically disseminates its own identity to its authenticated neighbors and continuously collects all other nodes' identities. Thus, nodes can build a secure map of other nodes for geographical routing. In routing, each node encrypts the packet by its key which is verified by the next hop en route. Such dissemination period was set to 30 s in this experiment. The routing of AO2P is similar to GPSR except

it has a contention phase in which the neighboring nodes of the current packet holder will contend to be the next hop. This contention phase is to classify nodes based on their distance from the destination node, and select a node in the class that is closest to destination. Contention can make the ad hoc channel accessible to a smaller number of nodes in order to decrease the possibility that adversaries participate, but concurrently this leads to an extra delay. Also, AO2P selects a position on the line connecting the source and destination that is further to the source node than the destination to provide destination anonymity, which may lead to long path length with higher routing cost than GPSR.

5.1 Network Models

We use two different network models, random way point model [17] and group mobility model [18]. With the random way point model as the default setting, we also compare the performance of ALERT in the group mobility model. In the group mobility model, we set the movement range of each group to 150 m with 10 groups [6] and to 200 m with five groups.

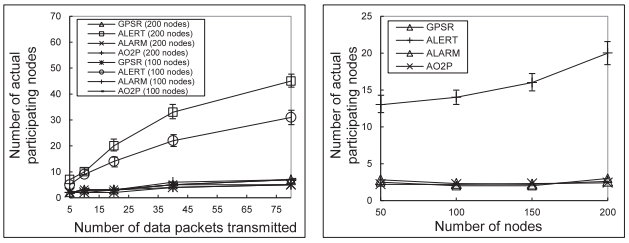
5.2 Parameters

The tests were carried out on NS-2.29 simulator using 802.11 as the MAC protocol with a standard wireless transmission range of 250 m and UDP/CBR traffic [31] with a packet size of 512 bytes. The test field in our experiment was set to a $1{,}000\ \mathrm{m} \times 1{,}000\ \mathrm{m}$ area with 200 nodes moving at a speed of 2 m/s, unless otherwise specified. The density was set to 50, 100, 150, and 200 nodes per square meters. The duration of each simulation was set to 100 s unless otherwise indicated. The number of pairs of S-D communication nodes was set to 10 and S-D pairs are randomly generated. S sends a packet to D at an interval of 2 s. The final results are the average of results of 30 runs. The confidence interval can be thus calculated from different runs and are shown when necessary. The confidence interval information is drawn along with the average point (in a "I" shape) on those figures.

For encryption, the symmetric encryption algorithm is AES and the public key encryption is RSA. Data are generated randomly according to the packet size specified in the paper. Packets are encrypted whenever needed. The encryption algorithm is single threaded, running along with other parts of the experiment on a 1.8 Ghz processor. A typical symmetric encryption costs several milliseconds while a public key encryption operation costs 2-3 hundred milliseconds.

We use the following metrics to evaluation the routing performance in terms of effectiveness on anonymity protection and efficiency:

1. *The number of actual participating nodes*. These nodes include *RF*s and relay nodes that actually participate in routing. This metric demonstrates the ability of ALERT's randomized routing to avoid routing pattern detection.
2. *The number of random forwarders*. This is the number of actual *RF*s in a S-D routing path. It shows routing anonymity and efficiency.
3. *The number of remaining nodes in a destination zone*. This is the number of original nodes remaining in a destination zone after a time period. A larger number provides higher anonymity protection to a destination and to counter the intersection attack. We measure this metric over time to show effectiveness on the destination anonymity protection.
4. *The number of hops per packet*. This is measured as the accumulated routing hop counts divided by the number of packets sent, which shows the efficiency of routing algorithms.
5. *Latency per packet*. This is the average time elapsed after a packet is sent and before it is received. It includes the time cost for routing and cryptography. This metric reflects the latency and efficiency of routing algorithms.
6. *Delivery rate*. This is measured by the fraction of packets that are successfully delivered to a destination. It shows the robustness of routing algorithms to adapt to mobile network environment.



(a) Different number of packets transmitted. (b) Different network size.

Fig. 10. The number of actual participating nodes.

5.3 The Number of Actual Participating Nodes

Fig. 10a demonstrates the cumulated actual participating nodes in ALERT, GPSR, ALARM, and AO2P, with 100 and 200 nodes moving at a speed of 2 m/s, respectively. Since ALARM and AO2P are similar to GPSR in the routing scheme and thus have similar number of actual participating nodes, we use GPSR to also represent ALARM and AO2P in discussing the performance difference between them and ALERT. We see that ALERT generates many more actual participating nodes since it produces many different routes between each S-D pair. The figure shows that the number of actual participating nodes is up to 30 in the 100 nodes case and is up to 45 in the 200 nodes case. The results are close to the analytical results of the number of possible participating nodes (approximately 30 and 60 in Fig. 8a). In ALERT, more nodes in the network produce more participating nodes because each routing involves more new random forwarders, which is a key property of ALERT to provide routing anonymity. On the contrary, GPSR only has slight increase in the number of participating nodes because it always takes the shortest path by greedy routing.

Fig. 10b shows the number of actual participating nodes after the transmission of 20 packets versus the number of nodes in the network. We see that the number of actual participating nodes in GPSR is steady with a marginal increase. This is due to the reason that the increased node density provides shorter routes. We also can see that ALERT generates dramatically more participating nodes than GPSR. GPSR has only 2-3 nodes while ALERT has 13-20. More participating nodes leads to more randomized routes

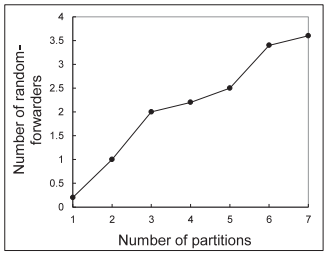


Fig. 11. The # of random forwarders.

that are difficult to detect or intercept. Therefore, the results in Figs. 10a and 10b illustrate higher route anonymity property of ALERT. On the contrary, the shortest routing paths in ALARM, AO2P, and GPSR follow the same greedy routing principle, which are easy to be identified by the adversaries through traffic analysis. Especially, when there are only few nodes communicate in the network, the route between two nodes could become very clear.

5.4 The Number of Random Forwarders

Fig. 11 demonstrates the number of *RF*s versus the number of partitions in ALERT. We see the average number of *RF*s follows approximately a linear trend as the number of partitions increases. This experimental result is consistent with the analytical results in Fig. 8b. A higher number of partitions H lead to more *RF*s, hence high anonymity protection. Recall that $H = log_2(\frac{\rho \cdot G}{k})$ and k controls the anonymity protection degree of the destination. Thus, k should be set to a value that will not generate a high cost for broadcasting while still providing high anonymity protection. Therefore, it is important to discover an optimal tradeoff point for H and k.

5.5 Destination Anonymity Protection

Fig. 12 depicts the number of remaining nodes with five partitions and a 2 m/s node moving speed when the node density equals 100, 150, and 200, respectively. The figure shows that the number of remaining nodes increases as node density grows while it decreases as time goes on. This is because higher node density leads to more nodes in the destination zone, and more nodes could remain in the destination zone after certain a time than with lower node density. Also, because of node mobility, the number of nodes that have moved out of the destination zone increases as time passes. This figure fits well with our analysis in Fig. 9a.

Fig. 13a shows the number of remaining nodes with different numbers of partitions H, and node moving speed, denoted by v. In this experiment, we set the node moving speed to 0, 2, and 4 m/s, respectively. We observe that

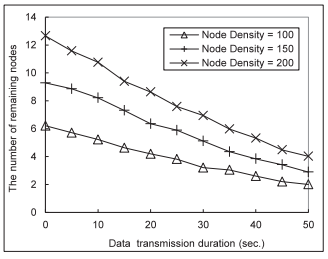
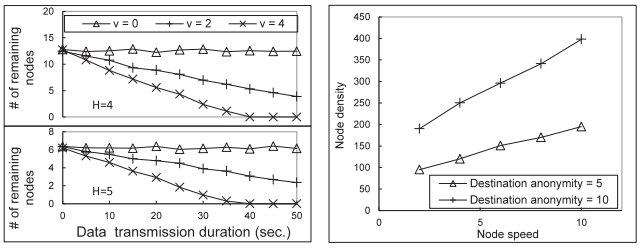


Fig. 12. Destination anonymity.



(a) H = 4 and 5 (200 nodes). (b) Fixed destination anonymity.

Fig. 13. Influence of node moving speed and partitions on destination anonymity.

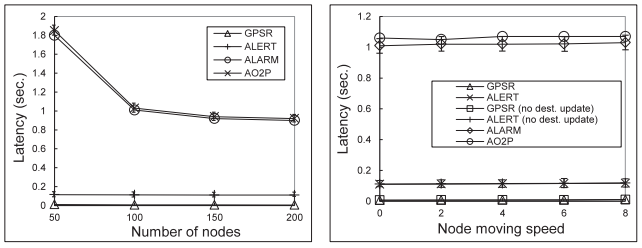
higher node mobility leads to less remaining nodes and hence negatively impacts the anonymity protection of the destination. This means ALERT is more suitable in low mobility environments to protect destinations from intersection attacks. This conclusion is consistent with the analytical results in Figs. 9a and 9b. Comparing the two figures in Fig. 13a, we observe similar slopes in both figures, which confirms the negative effect of node mobility on the destination anonymity protection. Further, the number of remaining nodes when $H = 4$ is more than that when $H = 5$. Less partitions generate larger area destination zone with more nodes, thus providing higher anonymity protection to the destination while also increasing the energy consumption in broadcasting.

In Fig. 13b, we fixed the number of nodes in destination zone (i.e., the number of remaining nodes in destination zone) and set the data transmission to 10 s. Therefore, destination anonymity is represented as a function of both node speed and density. We can see that as the node speed increases, the required node density also increases. This is reasonable because faster movement blanks out nodes originally in the destination zone more quickly.

5.6 Routing Performance

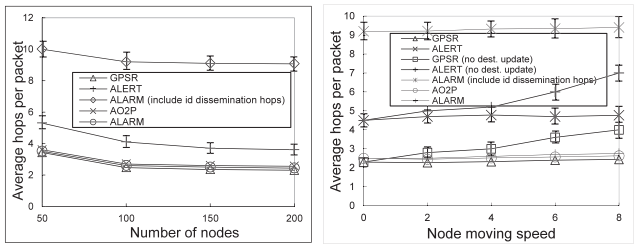
In this experiment, we evaluated the routing performance of ALERT compared with GPSR, AO2P, and ALARM in terms of latency, number of hops per packet, and delivery rate. We also conducted tests with and without destination update in location service to show the routing performance of different methods. In our experiment, for GPSR, if a destination node has moved away from its original position without location update, the forwarding nodes will continue to forward the packet to other nodes until the routing path length reaches a predefined TTL. The TTL was set to 10 in the experiments. In a transmission session, if the position of a packet's destination is changed but is not updated in the location service, the packet may not successfully reach the destination.

Fig. 14a presents the latency per packet versus the total number of nodes (i.e., node density). Recall that ALERT does not take the shortest path in routing, while ALARM and AO2P take the shortest path in routing. It is intriguing to see that the latency of ALERT is much lower than ALARM and AO2P. This is caused by the time cost of encryption. ALERT is based on symmetric key encryption for packets, which takes shorter time than the public key encryption used in ALARM and AO2P. Also, ALERT encrypts packets once, while AO2P needs to encrypt packets in each hop in routing and ALARM needs to



(a) Different node density. (b) Different node moving speed.

Fig. 14. Latency caused by encryption and routing.



(a) Different node density. (b) Different node moving speed.

Fig. 15. Transmission cost.

periodically authenticate neighbors. In ALARM and AO2P, the latency caused by the public key cryptography outweighs the benefit of short latency using the shortest path. Therefore, even though ALERT generates more routing hops than AO2P and ALARM as shown in Fig. 15, the latency of ALERT is still significantly lower than ALARM and AO2P. The results confirm that ALERT generates less cost due to encryption than ALARM and AO2P. The latency of AO2P is a little higher than ALARM because AO2P has a contention phase and may generate a slightly longer path length as explained previously.

We also see that ALERT generates a slightly longer latency than GPSR. ALERT does not aim to find a shortest route. Instead, it deliberately chooses a number of *RF*s to provide routing anonymity. Another observation is that the latency of all methods decreases as the node density increases. ALARM and AO2P exhibit a relatively faster drop, while ALERT's latency decreases from 12 to 11 ms and GPSR's latency decreases from 11 to 6 ms. This is because a higher node density provides more options for relay nodes, leading to shorter routing paths. Also, reduced public key encryption operations in ALARM and AO2P significantly reduce the latency. In ALERT, the transmission between two *RF*s depends on GPSR, so its latency is reduced as well.

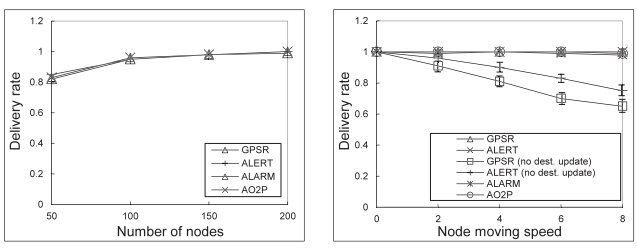
Fig. 14b shows the latency versus node moving speed varied from 2 to 8 m/s. We can also observe that AO2P generates marginally higher latency than ALARM, both of them produce dramatically higher latency than GPSR and ALERT, and ALERT produces slightly higher latency than GPSR due to the same reasons in Fig. 14a. When with destination update, experimental data indicate GPSR and ALERT have relatively stable latency with respect to node moving speed. This is because the destination node location can always be updated in time, so the routing path is always the shortest regardless of the moving speed. When without destination update, the experimental results shows that GPSR increases from 7 to 11 ms and ALERT increases from 11 to 12 ms though the phenomenon is not obvious in the figure. When a forwarding node fails to forward a message to the destination, it continues to forward the packet to other nodes until the path length reaches the $TTL = 10$. Thus, the number of hops in a route increases, leading to longer routing latency.

Fig. 15a shows the average hops per packet with the number of nodes. In order to show the high cost of the group key dissemination in ALARM, we also include the hops traversed for node identity dissemination into the routing hops for the metric calculation, denoted by "ALARM (include id dissemination hops)" in the figure. ALERT has approximately one more hop per packet than ALARM, AO2P, and GPSR since ALERT's random relay selection generates longer path length than the shortest path and others take the shortest path. The number of hops per packet of AO2P and ALARM is similar to GPSR because AO2P and ALARM use the GPSR routing mechanism. However, the GPSR routing algorithm with a strict node selection cannot provide anonymity since adversaries can easily observe the nodes in the routing path. The figure also shows that "ALARM (include id dissemination hops)" generates significantly higher hops per packet than others, which is doubled of that of ALERT. This result verifies the dramatically high cost of redundant traffic for anonymity in ALARM.

The periodical dissemination of node identities in ALARM costs an additional large number of hops. Combining the results in Fig. 14, we can conclude that compared to the hop-by-hop encryption AO2P method and the redundant traffic ALARM method, ALERT generates lower computing cost. Also, though ALERT leads to a slight increase in routing hop cost, it provides a higher anonymity guarantee due to its undetermined routing path.

Fig. 15b shows the average hops per packet when the moving speed of nodes is varied from 2 to 8 m/s. We see that the number of hops of ALERT and GPSR increases as the moving speed increases when there is no destination update. This is because the location change of nodes leads to longer route. When there is destination update, the destination of routing is always the updated location, so the packet will be routed to the destination following the shortest path regardless of the moving speed. We also see that "ALARM (include id dissemination hops)" still has the highest cost due to its periodical dissemination of node identity. ALERT has slightly higher hops per packet than ALARM, AO2P, and GPSR. These results are consistent with those in Fig. 15a due to the same reasons.

Fig. 16a shows the delivery rate versus the number of nodes with destination update. We see that delivery rate of all methods are close to 1, except in the sparse environment



(a) Different node density. (b) Different node moving speed.

Fig. 16. Transmission delivery rate.

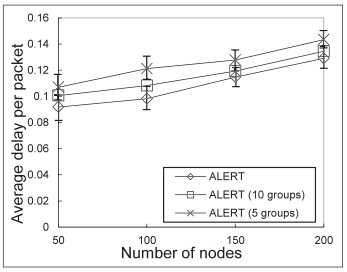


Fig. 17. Delay under different movement models.

where node density is only $50\ nodes/\mathrm{km}^2$. This is due to the unavailability of relay nodes in a sparse environment sometimes. In Fig. 16b, when there is destination update, ALERT and other methods can also maintain a delivery rate steadily with different node moving speeds from 2 to 8 m/s. For ALERT and the base-line method GPSR, when there is no destination update, the delivery rates decrease as node moving speed increases because of the mobility of destinations during data transmission. An interesting observation is that ALERT produces a higher delivery rate than GPSR. This is a benefit of the final local broadcast process in ALERT, which increases the possibility of packet delivery when the destination is not too far away.

We also measured the performance of ALERT under two movement models, random way point model [17] and group mobility model [18]. For group mobility model, we set the movement range of each group to 150 m with 10 groups and 200 m with five groups, respectively. Fig. 17 shows the delay of different movement models. It can be seen that the delay of ALERT increases slightly in the group movement model, this is because ALERT in the random way point model relies on the randomly distributed nodes around each sender and forwarders, while nodes are less randomly distributed in the group mobility model. As the number of groups increases, each group contains less nodes, and the mobility of the entire network will become more randomized. Therefore, ALERT with five groups generates higher delay than ALERT with 10 groups, as shown in the figure.

In summary, the experimental results exhibit consistency with the theoretical analysis and show that ALERT achieves better route anonymity protection compared with existing anonymous routing protocols due to its random relay node selection. It has significantly lower energy consumption compared to AO2P and ALARM, and provides comparable routing efficiency with AO2P, ALARM, and GPSR.

6 Related Work

Anonymous routing schemes in MANETs have been studied in recent years. By the different usage of topological information, they can be classified into on-demand or reactive routing methods [8], [33], [34], [32], [3], [4], [11], [10], [13], and proactive routing methods [5]. Also there are anonymous middleware working between network layer and application layer [9]. Since topology routing does not need the node location information, location anonymity protection is not necessary. Table 1 shows the classification of the methods along with their anonymity protection. To clearly show the featured anonymity protection in different reactive routing methods, the table provides a finer classification of different anonymity methods, including hop-by-hop encryption [8], [33], [34], [32], [3], [4], [11], [10] and redundant traffic routing [8], [11], [13].

In hop-by-hop encryption routing, a packet is encrypted in the transmission of two nodes en route, preventing adversaries from tampering or analyzing the packet contents to interrupt the communication or identify of the two communicating nodes. Hop-by-hop encryption routing can be further divided into onion routing and hop-by-hop authentication. In onion routing, packets are encrypted in the source node and decrypted layer by layer (i.e., hop by hop) along the routing path. It is used in Aad [8], ANODR [33] and Discount-ANODR [34] topological routing. Aad [8] combines onion routing, multicast, and uses packet coding policies to constantly change the packets in order to reinforce both destination and route anonymity. The onion used in ANODR [33] is called trapdoor boomerang onion (TBO), which uses a trapdoor function instead of public key-based encryption. ANODR needs onion construction in both route discovery and return routing, generating high cost. To deal with this problem, the authors further proposed Discount-ANODR that constructs onions only on the return routes.

Hop-by-hop authentication is used to prevent adversaries from participating in the routing to ensure route anonymity [32], [3], [4], [11], [10], [7], [35]. MASK [32] topological routing uses neighborhood authentication in routing path discovery to ensure that the discovered routes consist of legitimate nodes and are anonymous to attackers. The works in [3], [4], [11], [10] are based on geographic routing. In GSPR [3], nodes encrypt their location updates and send location updates to the location server. However, GSPR does not provide route anonymity because packets

TABLE 1
Summary of Existing Anonymous Routing Protocols

Category			Name	Identity anonymity	Location anonymity	Route anonymity
Reactive	Hop-by-hop encryption	Topology	MASK [32]	source	n/a	yes
			ANODR [33]	source, destination	n/a	yes
			Discount-ANODR [34]	source, destination	n/a	yes
		Geographic	Zhou *et al.* [3]	source, destination	source, destination	no
			Pathak *et al.* [4]	source, destination	source, destination	no
			AO2P [10]	source, destination	source, destination	no
			PRISM [6]	source, destination	source, destination	no
	Redundant traffic	Topology	Aad [8]	destination	n/a	yes
		Geographic	ASR [11]	source, destination	source, destination	no
			ZAP [13]	destination	destination	no
Proactive	Redundant traffic	Topology	ALARM [5]	source, destination	source	no
Middleware	Redundant traffic	Geographic	MAPCP [9]	source, destination	n/a	yes

always follow the shortest paths using geographic routing, and the route can be detected by adversaries in a long communication session. In [4], a mechanism called geographic hash is used for authentication between two hops en route, but the anonymity is compromised because the location of each node is know to nodes in the vicinity. In the AO2P [10] geographic routing algorithm, pseudonyms are used to protect nodes' real identities, and a node chooses the neighbor that can reduce the greatest distance from the destination. Since AO2P does not provide anonymity protection to destinations, the authors further improve it by avoiding the use of destination in deciding the classification of nodes. The improved AO2P selects a position on the line connecting the source and destination that is further to the source node than the destination and replaces the real destination with this position for distance calculation. ASR [11] conducts authentication between the source and the destination before data transmission. The source and each forwarder embed their public keys to the messages and locally broadcast the messages. The destination responds to the source in the same way. In each step, the response is encrypted using the previous node's public key so that only the previous forwarder can decrypt the message and further forward it. However, such public key dissemination in routing makes it possible for attackers to trace source/destination nodes. Ariadne [7] uses TESLA [36] to conduct broadcasting-style authentication between two neighboring hops en route. Although it uses symmetric key cryptography in the authentication, a high amount of traffic is inevitably incurred in broadcasting. SEAD [35] uses low-cost one-way hash functions rather than asymmetric cryptographic operations in conducting authentication for lower cost. However, all of these hop-by-hop encryption methods generate high cost due to the use of hop-by-hop public-key cryptography or complex symmetric key cryptography.

Redundant traffic-based routing uses redundant traffic, such as multicast, local broadcasting, and flooding, to obscure potential attackers. Multicast is used in the Aad [8] topological routing algorithm to construct a multicast tree or forest to hide the destination node. Broadcast is used in MAPCP topological routing [9] and other geographic routing protocols [5], [11]. ASR [11] shuffles packets to prevent traffic analysis in addition to the hop-by-hop authentication mentioned above. However, its routing anonymity is compromised because the public key dissemination in routing makes it possible for the attackers to trace back to the source and destination. ZAP [13] uses a destination zone, and locally broadcasts to a destination zone in order to reach the destination without leaking the destination identity or position. A disadvantage of redundant traffic-based methods is the very high overhead incurred by the redundant operations or packets, leading to high cost. Although some methods such as ZAP only perform local broadcast in a destination zone, these methods cannot provide source or routing anonymity.

ALARM [5] uses proactive routing, where each node broadcasts its location information to its authenticated neighbors so that each node can build a map for later anonymous route discovery. However, this map construction leaks destination node locations and compromises the route anonymity. Different from all other studied methods, MAPCP [9] is a middleware between network and application layers, in which every hop in the routing path executes probabilistic broadcasting that chooses a number of its neighbors with a certain probability to forward messages.

Mix zones [12] and GLS [24] are zone-based location services. Mix zones is an anonymous location service that unveils the positions of mobile users in a long time period in order to prevent users' movement from being tracked. Each location aware application that can monitor nodes' locations on top of Mix zones is only allowed to monitor the nodes that are registered to it. Therefore, by letting each node associate with some zones but stay unregistered, these users' location changes are untraceable in unregistered zones. Although GLS also uses hierarchical zone partitioning, its use is for location service while in ALERT, its use is for anonymous routing. ALERT is also different from GLS in the zone division scheme. A zone in ALERT is always divided into two smaller rectangles, while GLS divides the entire square area into four sub squares and then recursively divides these into smaller squares. The zone division in ALERT occurs when selecting a next forwarding node, so the zones are formed dynamically as a message is being forwarded. In contrast, the zone division and hierarchies in GLS are configured in advance and the location servers are selected based on the different hierarchies.

7 Conclusion and Future Work

Previous anonymous routing protocols, relying on either hop-by-hop encryption or redundant traffic, generate high cost. Also, some protocols are unable to provide complete source, destination, and route anonymity protection. ALERT is distinguished by its low cost and anonymity protection for sources, destinations, and routes. It uses dynamic hierarchical zone partitions and random relay node selections to make it difficult for an intruder to detect the two endpoints and nodes en route. A packet in ALERT includes the source and destination zones rather than their positions to provide anonymity protection to the source and the destination. ALERT further strengthens the anonymity protection of source and destination by hiding the data initiator/receiver among a number of data initiators/receivers. It has the "notify and go" mechanism for source anonymity, and uses local broadcasting for destination anonymity. In addition, ALERT has an efficient solution to counter intersection attacks. ALERT's ability to fight against timing attacks is also analyzed. Experiment results show that ALERT can offer high anonymity protection at a low cost when compared to other anonymity algorithms. It can also achieve comparable routing efficiency to the base-line GPSR algorithm. Like other anonymity routing algorithms, ALERT is not completely bulletproof to all attacks. Future work lies in reinforcing ALERT in an attempt to thwart stronger, active attackers and demonstrating comprehensive theoretical and simulation results.

Acknowledgments

This research was supported in part by US National Science Foundation grants CSR-1025649, OCI-1064230, CNS-1049947, CNS-1156875, CNS-0917056, CNS-1057530, CNS-1025652, CNS-0938189, CSR-2008826, and CSR-2008827,

Microsoft Research Faculty Fellowship 8300751, Sandia National Laboratories grant 10002282, and US Department of Energy's Oak Ridge National Laboratory including the Extreme Scale Systems Center located at ORNL and DoD 4000111689. An early version of this work [37] was presented in the Proceedings of ICPP 2009.

REFERENCES

[1] A. Pfitzmann, M. Hansen, T. Dresden, and U. Kiel, "Anonymity, Unlinkability, Unobservability, Pseudonymity, and Identity Management a Consolidated Proposal for Terminology, Version 0.31," technical report, 2005.

[2] Sk.Md.M. Rahman, M. Mambo, A. Inomata, and E. Okamoto, "An Anonymous On-Demand Position-Based Routing in Mobile Ad Hoc Networks," *Proc. Int'l Symp. Applications on Internet (SAINT),* 2006.

[3] Z. Zhi and Y.K. Choong, "Anonymizing Geographic Ad Hoc Routing for Preserving Location Privacy," *Proc. Third Int'l Workshop Mobile Distributed Computing (ICDCSW),* 2005.

[4] V. Pathak, D. Yao, and L. Iftode, "Securing Location Aware Services over VANET Using Geographical Secure Path Routing," *Proc. IEEE Int'l Conf. Vehicular Electronics and safety (ICVES),* 2008.

[5] K.E. Defrawy and G. Tsudik, "ALARM: Anonymous Location-Aided Routing in Suspicious MANETs," *Proc. IEEE Int'l Conf. Network Protocols (ICNP),* 2007.

[6] K.E. Defrawy and G. Tsudik, "PRISM: Privacy-Friendly Routing in Suspicious MANETs (and VANETs)," *Proc. IEEE Int'l Conf. Network Protocols (ICNP),* 2008.

[7] Y.-C. Hu, A. Perrig, and D.B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," *Wireless Networks,* vol. 11, pp. 21-38, 2005.

[8] I. Aad, C. Castelluccia, and J. Hubaux, "Packet Coding for Strong Anonymity in Ad Hoc Networks," *Proc. Securecomm and Workshops,* 2006.

[9] C.-C. Chou, D.S.L. Wei, C.-C. Jay Kuo, and K. Naik, "An Efficient Anonymous Communication Protocol for Peer-to-Peer Applications over Mobile Ad-Hoc Networks," *IEEE J. Selected Areas in Comm.,* vol. 25, no. 1, pp. 192-203, Jan. 2007.

[10] X. Wu, "AO2P: Ad Hoc On-Demand Position-Based Private Routing Protocol," *IEEE Trans. Mobile Computing,* vol. 4, no. 4, pp. 335-348, July/Aug. 2005.

[11] B. Zhu, Z. Wan, M.S. Kankanhalli, F. Bao, and R.H. Deng, "Anonymous Secure Routing in Mobile Ad-Hoc Networks," *Proc. IEEE 29th Ann. Int'l Conf. Local Computer Networks (LCN),* 2004.

[12] A.R. Beresford and F. Stajano, "Mix Zones: User Privacy in Location-Aware Services," *Proc. IEEE Second Ann. Conf. Pervasive Computing and Comm. Workshops (PERCOMW),* 2004.

[13] X. Wu, J. Liu, X. Hong, and E. Bertino, "Anonymous Geo-Forwarding in MANETs through Location Cloaking," *IEEE Trans. Parallel and Distributed Systems,* vol. 19, no. 10, pp. 1297-1309, Oct. 2008.

[14] K. El-Khatib, L. Korba, R. Song, and G. Yee, "Anonymous Secure Routing in Mobile Ad-Hoc Networks," *Proc. Int'l Conf. Parallel Processing Workshops (ICPPW),* 2003.

[15] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu, "Data-Centric Storage in Sensornets with GHT, a Geographic Hash Table," *Mobile Network Applications,* vol. 8, no. 4, pp. 427-442, 2003.

[16] J. Raymond, "Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems," *Proc. Int'l Workshop Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability (WDIAU),* pp. 10-29, 2001.

[17] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communications and Mobile Computing,* vol. 2, pp. 483-502, 2002.

[18] X. Hong, M. Gerla, G. Pei, and C.C. Chiang, "A Group Mobility Model for Ad Hoc Wireless Networks," *Proc. Second ACM Int'l Workshop Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM),* 1999.

[19] Debian Administration, http://www.debian-administration.org/users/dkg/weblog/48, 2012.

[20] X. Wu, "DISPOSER: Distributed Secure Position Service in Mobile Ad Hoc Networks: Research Articles," *Wireless Comm. and Mobile Computing,* vol. 6, pp. 357-373, 2006.

[21] M.F. Mokbel, C.-Y. Chow, and W.G. Aref, "The New Casper: Query Processing for Location Services without Compromising Privacy," *Proc. 32nd Int'l Conf. Very Large Databases (VLDB),* 2006.

[22] J. Li, J. Jannotti, D.S.J. De, D.S.J. De Couto, D.R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," *Proc. ACM MobiCom,* 2000.

[23] Y. Xue, B. Li, and K. Nahrstedt, "A Scalable Location Management Scheme in Mobile Ad-Hoc Networks," technical report, 2001.

[24] J. Li, J. Jannotti, D.S.J. De, C. David, R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," *Proc. ACM MobiCom,* 2000.

[25] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," *Int'l J. Uncertainity Fuzziness Knowledge-Based Systems,* vol. 10, no. 5, pp. 557-570, 2002.

[26] N.R. Potlapally, S. Ravi, A. Raghunathan, and N.K. Jha, "Analyzing the Energy Consumption of Security Protocols," *Proc. Int'l Symp. Low Power Electronics and Design (ISLPED),* 2003.

[27] H. Frey and I. Stojmenovic, "On Delivery Guarantees of Face and Combined Greedy-Face Routing in Ad Hoc and Sensor Networks," *Proc. ACM MobiCom,* 2006.

[28] K.C. Lee, J. Haerri, L. Uichin, and M. Gerla, "Enhanced Perimeter Routing for Geographic Forwarding Protocols in Urban Vehicular Scenarios," *Proc. IEEE GlobeCom Workshops,* 2007.

[29] X. Wu, "DISPOSER: Distributed Secure Position Service in Mobile Ad Hoc Networks: Research Articles," *Wireless Comm. and Mobile Computing,* vol. 6, pp. 357-373, 2006.

[30] "Ke Liu's NS2 Code," http://www.cs.binghamton.edu/~kliu/research/ns2code/index.html, 2012.

[31] "The Network Simulator - ns-2," http://www.isi.edu/nsnam/ns, 2012.

[32] Y. Zhang, W. Liu, and W. Luo, "Anonymous Communications in Mobile Ad Hoc Networks," *Proc. IEEE INFOCOM,* 2005.

[33] J. Kong, X. Hong, and M. Gerla, "ANODR: Anonymous on Demand Routing Protocol with Untraceable Routes for Mobile Ad-Hoc Networks," *Proc. ACM MobiHoc,* pp. 291-302, 2003.

[34] L. Yang, M. Jakobsson, and S. Wetzel, "Discount Anonymous On Demand Routing for Mobile Ad Hoc Networks," *Proc. Securecomm and Workshops,* 2006.

[35] Y.-C. Hu, D.B. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," *Proc. IEEE Workshop Mobile Computing Systems and Applications (WMCSA),* 2002.

[36] A. Perrig, R. Canetti, D. Song, and J.D. Tygar, "Efficient and Secure Source Authentication for Multicast," *Proc. Network and Distributed System Security Symp. (NDSS),* 2001.

[37] L. Zhao and H. Shen, "ALERT: An Anonymous Location-Based Efficient Routing Protocol in MANETs," *Proc. Int'l Conf. Parallel Processing (ICPP),* 2011.

Haiying Shen received the BS degree in computer science and engineering from Tongji University, China, in 2000, and the MS and PhD degrees in computer engineering from Wayne State University in 2004 and 2006, respectively. She is currently an assistant professor in the Electrical and Computer Engineering Department at Clemson University. Her research interests include distributed computer systems and computer networks with an emphasis on P2P and content delivery networks, mobile computing, wireless sensor networks, and grid and cloud computing. She is a member of the IEEE.

Lianyu Zhao received the BS and MS degrees in computer science from Jilin University, China, and is currently working toward the PhD degree in the Electrical and Computer Engineering Department at Clemson University. His research interests include wireless networks, routing protocols, applications, and security issues in P2P networks. He is a student member of the IEEE.