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

Lianyu Zhao and Haiying Shen
Department of Electrical and Computer Engineering
Clemson University Clemson, SC 29634
{*lianyuz, shenh*}*@clemson.edu*

***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. 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 non-traceable anonymous route. In addition, it hides the data initiator/receiver among many initiators/receivers. Thus, ALERT offers anonymity protection to sources, destinations, and routes. It also has strategies to effectively counter intersection and timing attacks. Experimental results 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.**

I. Introduction

Mobile Ad Hoc Networks (MANETs) feature self-organizing and independent infrastructures, which make them an ideal choice for military uses such as communication and information sharing in battlefields. However, the innate on-air nature of MANETs makes them 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, 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. For identity and location anonymity of sources and destinations, no one else knows the real identities and exact locations of the sources and destinations except themselves. 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 [21]), 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 [11], [13], [19], [23], [35] and redundant traffic [4], [5], [9], [16], [29], [32], [36]. Since public-key based encryption and high traffic generate significantly high cost, most of the current approaches are limited by focusing on enforcing anonymity at a heavy cost to precious resources. In addition, many approaches cannot provide all of the aforementioned anonymity protections. For example, ALARM [13] cannot protect the location anonymity of source and destination, SDDR [14] cannot provide route anonymity, and ZAP [32] only focuses on destination anonymity. Many anonymity routing algorithms [11], [13], [19], [29], [32], [35], [36] are based on a geographic routing protocol [24] which 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.

In order to provide high anonymity protection 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 non-traceable 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 Greedy Perimeter Stateless Routing (GPSR) [24] 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 [25] and timing attacks [25].

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 [25]. ALERT can also avoid timing attacks because of its non-fixed routing paths for a source-destination pair.
4) *Extensive simulations.* We conducted comprehensive experiments to evaluate its performance in comparison with other anonymous protocols.

The remainder of this paper is organized as follows. In Section II, we describe related anonymous routing approaches in MANETs. In Section III, we present the design of the ALERT routing protocol. Experimental performance of the ALERT protocol is evaluated in Section IV. The conclusion and future work are given in Section V.

II. Related Work

Anonymous routing has been studied for decades in traditional wired environments. Approaches such as Ants [6], Mix [7], broadcast [28], crowds [27], Onion [26] and its second generation TOR [12], along with the Freenet [10] file sharing system have been proposed to provide anonymity [8].

Anonymous routing schemes in MANETs have been studied in recent years. They can be classified into two categories: hop-by-hop encryption routing [4], [17], [19], [29], [33]–[36] and redundant traffic-based routing [4], [9], [13], [32], [36]. Table I shows the classification of the methods along with their anonymity protection. Since topology routing does not need the node location information, location anonymity protection is not necessary.

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 [4], ANODR [17] and Discount-ANODR [33] topological routing. Aad [4] combines onion routing and 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 [17] is called TBO (trapdoor boomerang onion), 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 [15], [16], [19], [29], [34]–[36]. MASK [34] 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 [19], [29], [35], [36] are based on geographic routing. In GSPR [35], nodes encrypt their location updates and send location updates to the location server. However, GSPR does not provide route anonymity because packets always follow the shortest paths using geographic routing and the route can be detected by adversaries in a long communication session. In [19], 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 [29] 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 [36] conducts authentication between the source and the destination before data transmission. The source and each forwarder embeds 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 [16] uses TESLA [20] 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 [15] 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 [4] topological routing algorithm to construct a multicast tree or forest to hide the destination node. Broadcast is used in

Table I: Summary of existing anonymous routing protocols.

Category		Name	Identity anonymity	Location anonymity	Route anonymity
Hop-by-hop encryption	Topology routing	MASK [34]	source	n/a	yes
		ANODR [17]	source, destination	n/a	yes
		Discount-ANODR [33]	source, destination	n/a	yes
	Geographic routing	Zhou *et al.* [35]	source, destination	source, destination	no
		Pathak *et al.* [19]	source, destination	source, destination	no
		AO2P [29]	source, destination	source, destination	no
		PRISM [11]	source, destination	source, destination	no
Redundant traffic	Topology routing	Aad [4]	destination	n/a	yes
		MAPCP [9]	source, destination	n/a	yes
	Geographic routing	ALARM [13]	source, destination	source	no
		ASR [36]	source, destination	source, destination	no
		ZAP [32]	destination	destination	no

MAPCP topological routing [9] and other geographic routing protocols [13], [36]. Specifically, in MAPCP [9], every hop in the routing path executes probabilistic broadcasting that chooses a number of its neighbors with a certain probability to forward messages. In ALARM [13], 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. ASR [36] 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 [32] 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. Also, although some methods such as ZAP only perform local broadcast in a destination zone, these methods cannot provide source or routing anonymity.

Mix zones [5] and GLS [18] 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' locations change is 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.

III. ALERT: An Anonymous Location-based Efficient Routing Protocol

A. Attack model

In this work, attackers can be battery powered sensors 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.

(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 is secure to a certain degree when the key is not known to the attackers.

B. 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

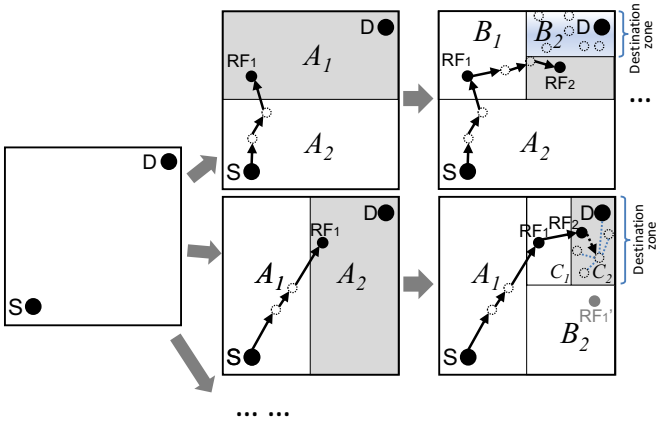


Figure 1: Examples of different zone partitions.

in the network. To avoid pseudonym collision, we use a collision-resistant hash function, such as SHA-1 [2], to hash a node's MAC address and current time stamp. A node's pseudonym expires after a specific time period in order to avoid adversaries associating the pseudonyms with nodes. 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.

ALERT uses the DISPOSER location service [30] to enable each source node to securely obtain the location and the public key of the destination. The public key is used to enable two nodes to securely establish a symmetric key K_s for secure communication. For example, source node A sends the location request containing destination B's identity to the service. Then the location service returns an encrypted position and the public key of B, which can be decrypted by A using the pre-distributed shared key between A and its location service.

C. Design of 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 Figure 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 to 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.

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. Specifically, in the ALERT routing, each data source or forwarder partitions the zone it resides in order to separate itself and Z_D into two zones. 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*). In the last step, the data is broadcasted to k nodes in Z_D, providing k-anonymity to the destination.

Zone position refers to the upper-left and bottom-right coordinates of a zone. One problem is finding the position of 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

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

where G is the size of the entire network area. Using the calculated H, the size G and position $(0,0),(x_G,y_G)$ of the entire network area, and the position of D, 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 resulting 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 $(0,0),(4,2)$, if $H = 3$ and the destination position is $(0.5,0.8)$, the resulting destination zone position is $(0,0),(1,1)$ and its size is $\frac{8}{2^3} = 1$.

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 H^{th} partitioned zone. Each packet forwarder needs this position to check whether it is separated from the destination after a partition and whether it resides in Z_D. (2) The encrypted zone position of the H^{th} partitioned zone of S using D's public key, which is the destination for data response. (3) The randomly selected TD for routing to the next *RF*. And (4) A bit (i.e., 0/1), which is flipped by each *RF*, indicating the partition direction (horizontal or vertical) of the next *RF*. 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 [22] 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, encrypted using D's public key, into a packet. Later, D sends S its requested contents, encrypted with K_s, decrypted by its own public key. Therefore, the packets communicated between S and D can be efficiently and securely protected using K_s.

Figure 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. The reason we use Z_D rather than D is to avoid exposure of 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. 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 re-send the packets.

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. The lower part of Figure 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.

ALERT contributes to achieving 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 secretly 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 second "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 sending 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.

D. Anonymity properties and resilience to timing attacks

ALERT offers identity and location anonymity of the source and destination, as well as route anonymity. Unlike geographic routing [19], [29], [31], [35], [36] 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 makes it difficult for an intruder to observe a statistical pattern of transmission. 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 [21]. 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

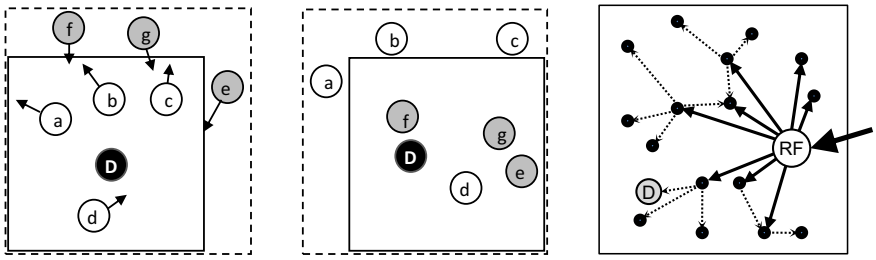


Figure 2: Intersection attack and solution.

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.

In timing attacks [25], 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 five 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.

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.

E. 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 [25]. 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.

Figure 2(a) 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. Figure 2(b) 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 [32] 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 2(c) 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.

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.

IV. Performance Evaluation

In this section, we provide an 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. 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 [29] and ALARM [13], 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 [1] in the experiments. 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 with its key, which is verified by the next hop en route. This dissemination period was set to 30s 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. 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 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 a long path length with a higher routing cost than GPSR.

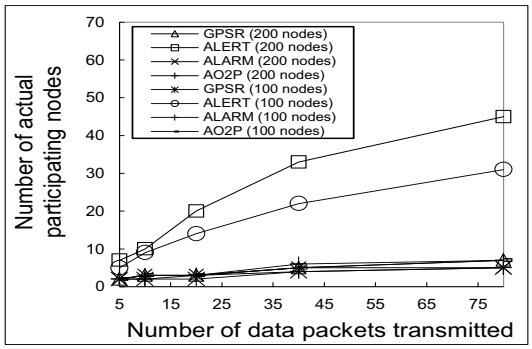
The tests were carried out on an NS-2.29 simulator using 802.11 as the MAC protocol with a standard wireless transmission range of 250m and UDP/CBR traffic [3] with a packet size of 512 bytes. The test field in our experiment was set to a $1000m \times 1000m$ area with 200 nodes moving at a speed of 2m/s, unless otherwise specified. 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 2s. The final results are the average of results of 5 runs. We use the following metrics:

(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 the performance in routing anonymity and efficiency.
(3) *The number of remanent 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.
(4) *The number of hops per packet*. This is calculated as the accumulated routing hop count 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.

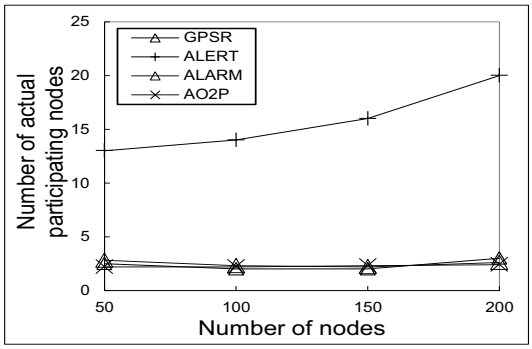
A. The number of actual participating nodes

Figure 3(a) demonstrates the cumulated actual participating nodes in ALERT, GPSR, ALARM and AO2P, with 100 and 200 nodes moving at a speed of 2m/s. Since ALARM, GPSR and AO2P have a similar 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 he number of actual participating nodes up to 30 in the 100 nodes case and is up to 45 in the 200 nodes case. In ALERT, more nodes in the network produce more actual participating nodes because each routing involves different random forwarders, which is a key property of ALERT to provide routing anonymity. On the contrary, GPSR only has a slight increase in the number of participating nodes because it always takes the shortest path based on greedy routing.

Figure 3(b) 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 because the increased node density provides shorter routes. We can also 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 that are difficult to detect or intercept. Therefore, the results in Figure 3(a) and Figure 3(b) illustrate the higher route anonymity property of ALERT. On the contrary, the shortest routing paths in ALARM, AO2P and GPSR follow the same greedy routing



(a) Different number of packets transmitted.



(b) Different network size.

Figure 3: The number of actual participating nodes.

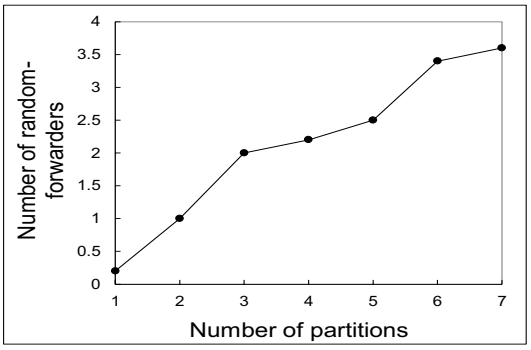


Figure 4: The # of random-forwarders.

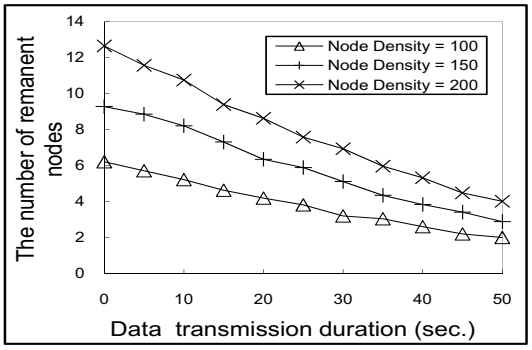


Figure 5: Destination anonymity.

principle, which are easy to identify by adversaries through traffic analysis. Especially, when there are only few nodes that communicate in the network, the route between two nodes could become very clear.

B. The number of random-forwarders

Figure 4 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. A higher number of partitions H leads to more *RF*s, hence higher 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.

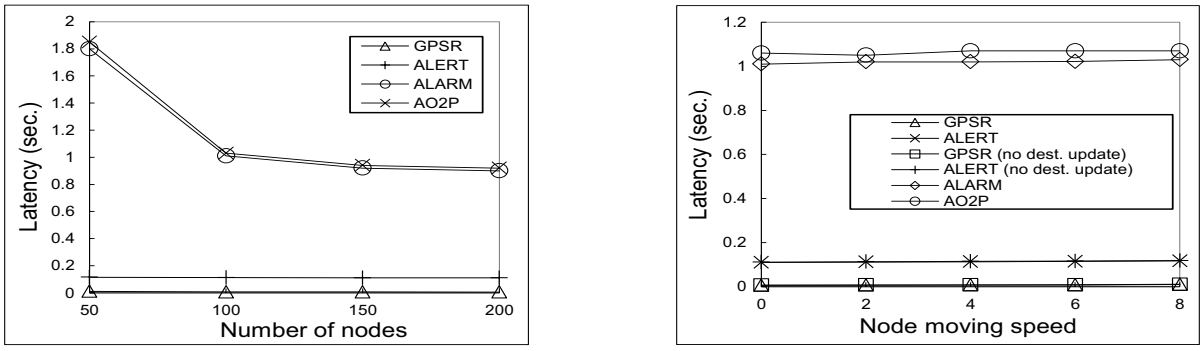
C. Destination anonymity protection

Figure 5 depicts the number of remanent nodes with 5 partitions and a 2m/s node moving speed when the node density equals 100, 150, and 200. The figure shows that the number of remanent nodes increases as node density grows while it decreases as time goes on. This is because a higher node density leads to more nodes in the destination zone and a greater chance that more nodes remain in the destination zone after a certain time. Also, because of node mobility, the number of nodes that have moved out of the destination zone increases as time passes.

D. 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 a 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.

Figure 6(a) 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.



(a) Different node density.

(b) Different node moving speed.

Figure 6: Latency caused by encryption and routing.

ALERT is based on symmetric key encryption for packets, which takes a shorter time than the public key encryption used in ALARM and AO2P. Also, ALERT encrypts packet once, while AO2P needs to encrypt packet in each hop in routing and ALARM needs to periodically authenticate neighbors. 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 12ms to 11ms and GPSR's latency decreases from 11ms to 6ms. 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.

Figure 6(b) shows the latency versus node moving speed varied from 2m/s to 8m/s. We can also observe that AO2P generates marginally higher latency than ALARM, both of those produce dramatically higher latency than GPSR and ALERT, and ALERT produces slightly higher latency than GPSR for the same reasons as in Figure 6(a). Experimental data indicates GPSR and ALERT have relatively stable latency with respect to node moving speed with destination update. This is because the destination node location can always be timely updated, so the routing path is always the shortest regardless of the moving speed. When without destination update, the experimental results show that GPSR increases from 7ms to 11ms and ALERT increases from 11ms to 12 ms, though this 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.

V. 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. 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 bullet-proof to all attacks. Future work lies in reinforcing ALERT in an attempt to thwart stronger, and active attackers and demonstrating comprehensive theoretical and simulation results.

Acknowledgements

This research was supported in part by U.S. NSF grants OCI-1064230, CNS-1049947, CNS-1025652, CNS-1025649, CNS-1057530 and CNS-0917056, Microsoft Research Faculty Fellowship 8300751, and Sandia National Laboratories grant 10002282.

References

[1] http://www.cs.binghamton.edu/~kliu/research/ns2code/index.html.

[2] http://www.debian-administration.org/users/dkg/weblog/48.

[3] http://www.isi.edu/nsnam/ns/.

[4] I. Aad, C. Castelluccia, and J. Hubaux. Packet coding for strong anonymity in ad hoc networks. In *Proc. of Securecomm*, 2006.

[5] A. R. Beresford and F. Stajano. Mix zones: User privacy in location-aware services. In *Proc. of PERCOMW*, 2004.

[6] I. Bouazizi. ARA - The Ant-Colony Based Routing Algorithm for MANETs. In *Proc. of ICPPW*, 2002.

[7] D. Chaum, C. O. T. Acm, R. Rivest, and D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24:84–88, 1981.

[8] T. Chothia and K. Chatzikokolakis. A Survey of Anonymous Peer-to-Peer File-Sharing. In *Proc. of NCUS*, pages 744–755, 2005.

[9] C.-C. Chou, D. S. Wei, C.-C. J. Kuo, and K. Naik. An efficient anonymous communication protocol for peer-to-peer applications over mobile ad-hoc networks. In *JSAC*, pages 192–203, 2007.

[10] I. Clarke, O. S, O. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *LNCS*, 2001.

[11] K. E. Defrawy and G. Tsudik. Prism: Privacy-friendly routing in suspicious manets (and vanets). In *Proc. of ICNP*, 2008.

[12] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proc. of the USENIX Security*, pages 303–320, 2004.

[13] K. El Defrawy and G. Tsudik. Alarm: Anonymous location-aided routing in suspicious manets. In *Proc. of ICNP*, 2007.

[14] K. El-Khatib, L. Korba, R. Song, and G. Yee. Anonymous secure routing in mobile ad-hoc networks. In *Proc. of ICPPW*, 2003.

[15] Y.-C. Hu, D. B. Johnson, and A. Perrig. SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks. In *Proc. of WMCSA*, 2002.

[16] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: a secure on-demand routing protocol for ad hoc networks. *Wirel. Netw.*, 11:21–38, 2005.

[17] J. Kong, X. Hong, and M. Gerla. ANODR: Anonymous on demand routing protocol with untraceable routes for mobile ad-hoc networks. In *Proc. of MobiHoc*, pages 291–302, 2003.

[18] J. Li, J. Jannotti, D. S. J. De, C. David, R. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In *Proc. of MOBICOM*, pages 120–130, 2000.

[19] V. Pathak, Y. Danfeng, and L. Iftode. Securing location aware services over VANET using geographical secure path routing. In *Proc. of ICVES*, 2008.

[20] A. Perrig, R. Canetti, D. Song, and J. D. Tygar. Efficient and secure source authentication for multicast. In *Proc. of NDSS*, 2001.

[21] 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.

[22] N. R. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha. Analyzing the energy consumption of security protocols. In *Proc. of ISLPED '03*, pages 30–35, 2003.

[23] S. M. M. Rahman, M. Mambo, A. Inomata, and E. Okamoto. An Anonymous On-Demand Position-Based Routing in Mobile Ad Hoc Networks. In *Proc. of SAINT*, pages 300–306, 2006.

[24] 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. *Mob. Netw. Appl.*, 8(4):427–442, 2003.

[25] J. Raymond. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In *Proc. of WDIAU*, pages 10–29, 2001.

[26] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous connections and onion routing. *IEEE JSAC*, 16:482–494, 1998.

[27] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for web transactions. *ACM TISS*, 1, 1998.

[28] R. Sherwood, B. Bhattacharjee, and A. Srinivasan. P5 : A Protocol for Scalable Anonymous Communication. In *Proc. of IEEE S&P*, 2002.

[29] X. Wu. AO2P: Ad Hoc On-Demand Position-Based Private Routing Protocol. *IEEE TMC*, 2005.

[30] X. Wu. DISPOSER: distributed secure position service in mobile ad hoc networks: Research Articles. *WCMC*, 6(3):357–373, 2006.

[31] X. Wu. Disposer: distributed secure position service in mobile ad hoc networks: Research articles. *Wirel. Commun. Mob. Comput.*, 6(3), 2006.

[32] X. Wu, J. Liu, X. Hong, and E. Bertino. Anonymous Geo-Forwarding in MANETs through Location Cloaking. *IEEE TPDS*, 2008.

[33] L. Yang, M. Jakobsson, and S. Wetzel. Discount Anonymous On Demand Routing for Mobile Ad hoc Networks. In *Proc. of Securecomm*, 2006.

[34] Y. Zhang, W. Liu, and W. Luo. Anonymous Communications in Mobile Ad Hoc Networks. In *Proc. of INFOCOM*, 2005.

[35] Z. Zhi and Y. K. Choong. Anonymizing Geographic Ad Hoc Routing for Preserving Location Privacy. In *Proc. of ICDCSW*, pages 646–651, 2005.

[36] B. Zhu, Z. Wan, M. S. Kankanhalli, F. Bao, and R. H. Deng. Anonymous Secure Routing in Mobile Ad-Hoc Networks. In *Proc. of LCN*, 2004.