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 $\frac{G}{2^H}$ 2H. 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.

3.2 Parameters

The following metrics are used to evaluation the routing Performance in terms of effectiveness on anonymity protection and efficiency:

[1] *The number of actual participating nodes.* These nodes include RFs 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 RFs 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.

[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.

3.3 Packet Format of AZPR

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

[1] The destination zone position Z_D, i.e., the H^{th} partitioned zone.

[2] The encrypted zone position of the H^{th} partitioned zone of *S* using *D*'s public key.

[3] The current randomly selected temporary destination (*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*.

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 calculates the values. In order to hide the packet content from adversaries, AZPR 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, AZPR uses symmetric key encryption for transmitted data. Recall that *S* can get *D*'s public key from the secure location service. In an *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 $K^S{}_s$.

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)

Figure 4: Packet format of AZPR

Fig. 4 shows the packet format of AZPR, which omits the MAC header. Because of the randomized routing nature in AZPR, we have a universal format for RREQ/RREP/NAK. Anode 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 [7] 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 H^{th} 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} pub is used for the protection of source anonymity and ,(Bitmap)K^{RN}_{pub} pub is used for solving intersection attack. When node *A* wants to know the location and public key of another node B, it will contact its location server, thus there is no need to exchange shared keys between nodes.

3.4 Source Anonymity

AZPR 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 ϵ $[t, t+t_0]$ before sending out

Licensed Under Creative Commons Attribution CC BY

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*. Notify and go provides difficult for an attacker to analyze traffic to discover *S* even if it receives the first notification.

4. Anonymity Protection and Strategies against Attack

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

4.1 Anonymity Protection

AZPR offers identity and location anonymity of the source and destination, as well as route anonymity. AZPR makes the route between *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 *RFs* 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. Also, the anonymous path between *S* and *D* ensures that nodes on the path do not know where the endpoints are. AZPR incorporates the "notify and go" mechanism to prevent an intruder from identifying which node within the source neighborhood has initiated packets. AZPR 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 AZPR prevents an intruder from intercepting packets or compromising vulnerable nodes en route to issue DoS attacks. In AZPR, 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.

4.2 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.

Avoiding the exhibition of interaction between communication nodes is a way to counter timing attacks. In AZPR, 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 changes constantly, which again keeps an intruder from identifying the *S* and *D*.

4.3. 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 AZPR 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

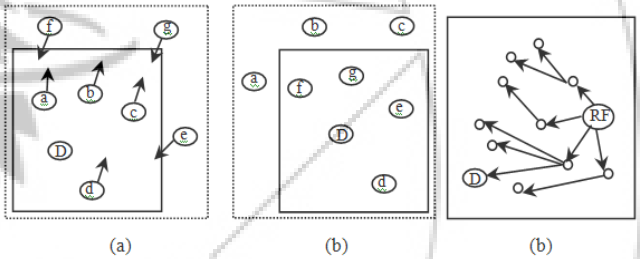

Figure 5: Intersection Attack and Solutions

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. 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 *pkt1* 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 *pkt2*. 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 *pkt1* and *pkt2* are mixed, an attacker observes that *D* is not in the recipient set of *pkt1* though *D* receives

Licensed Under Creative Commons Attribution CC BY

pkt1 in the delivery time of *pkt2*. 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.

5. Simulation Result

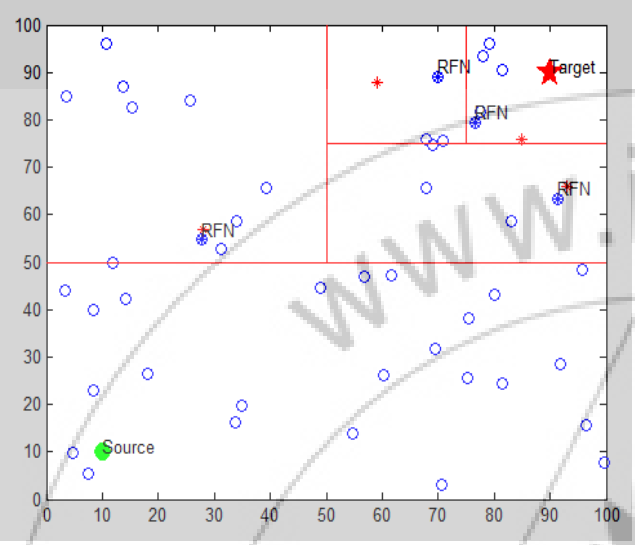

Figure 6: Result demonstrating a network partitions along with Random Forwarders

The tests were carried out on MATLAB simulator using MAC protocol with a standard wireless transmission range of 250 m and UDP/CBR traffic with a packet size of 512 bytes. The test field in our experiment was set to a 100m * 100m area with 50 nodes moving at a speed of 2 m/s, unless otherwise specified. The density was set to 5 to 10 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 2m/s. The final results are the average of results of 30 runs.

6. Conclusion

The anonymous routing protocols in MANET relying on either hop-by-hop encryption or redundant traffic generate high cost and cannot provide complete anonymity. To provide anonymity protection for source, destination and route at low cost, an algorithm AZPR is introduced in this paper. This algorithm partition the network into zones and selection of random forwarder provides route anonymity and consumes less resource. Although the randomized routing provides anonymity for the source and destination, the Notify and Go mechanism that we have introduced strengthen the source anonymity. The two tier broadcasting mechanism used to forward the packets to all the nodes in the destination zone (D_Z), strengthen the destination anonymity. These two mechanisms provide resiliency to intersection and timing attacks. Like other anonymity routing algorithms, AZPR is not completely resistant to all attacks. Future work lies in reinforcing AZPR in an attempt to thwart stronger, active attackers and demonstrating comprehensive theoretical and simulation results.

References

[1] K.E. Defrawy and G. Tsudik, "ALARM: Anonymous Location-Aided Routing in Suspicious MANETs," *Proc. IEEE Int'l Conf. Network Protocols (ICNP),* 2007.
[2] K.E. Defrawy and G. Tsudik, "PRISM: Privacy-Friendly Routing in Suspicious MANETs (and VANETs)," *Proc. IEEE Int'l Conf. etwork Protocols (ICNP)*, 2008.
[3] 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.
[4] Aad, C. Castelluccia, and J. Hubaux, "Packet Coding for Strong Anonymity in Ad Hoc Networks," *Proc. Securecomm and Workshops*, 2006.
[5] 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.
[6] 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.
[7] 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.
[8] 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.
[9] 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.
[10] Y. Zhang, W. Liu, and W. Luo, "Anonymous Communications in Mobile Ad Hoc Networks," *Proc. IEEE INFOCOM,* 2005.
[11] 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.
[12] 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.
[13] L. Zhao and H. Shen, "ALERT: An Anonymous Location-Based Efficient Routing Protocol in MANETs," *Proc. Int'l Conf. Parallel Processing (ICPP),* 2011.

Author Profile

Ayeesha Siddiqha received the B.E degree in Computer Science and Engineering from Visvesvaraya Technological University, Belgaum in 2011 and M.Tech degree Computer Science and Engineering from Visvesvaraya Technological University, Belgaum in 2013. At Present she is working as an Assistant Professor in Computer Science Department in Malnad College of Engineering, Hassan

Licensed Under Creative Commons Attribution CC BY

Arshad Khan received the B.E degree in Computer Science and Engineering from Visvesvaraya Technological University, Belgaum in 2011 and persuing M.Tech degree Computer Science and Engineering from Visvesvaraya Technological University, Belgaum.

www.ijsr.net
IJSR
ISSN (Online): 2319-7064

Licensed Under Creative Commons Attribution CC BY