Anonymizing Geographic Ad Hoc Routing for Preserving Location Privacy

Zhi Zhou and Kin Choong Yow

(Corresponding author: Zhi Zhou)

School of Computer Engineering, Nanyang Technological University
Singapore 639798 (Email: zhouzhi@pmail.ntu.edu.sg, kcyow@ntu.edu.sg)

(Received Aug. 18, 2005; revised and accepted Oct. 4, 2005)

Abstract

Due to the utilization of location information, geographic ad hoc routing presents superiority in scalability compared with traditional topology-based routing in mobile ad hoc networks. However, the consequent solicitation for location presence incurs severe concerns of location privacy, which has not been properly studied. In this paper, we attempt to preserve location privacy based on the idea of dissociating users' location information with its identity. We propose an anonymous geographic routing algorithm which includes three components to avoid the explicit exposure of identity and location in communication without compromising the efficiency guaranteed by geographic routing.

Keywords: Ad hoc, anonymity, geographic routing, security

1 Introduction

Geographic routing [1, 3, 8, 11, 19] is an important class of ad hoc routing protocols compared with topology-based schemes [7, 14]. Rather than discovering the topology for finding routes, routing decisions in geographic routing are made by measuring the geographic superiority among neighbors. The attempt to utilize available location information helps making localized decisions that are essential to the network scalability [18].

While the location usage promises enormous benefits in terms of protocol functionality, it is not practical for this class of routing protocols to be further applied into real life environments. Most of the prior works assume a privacy-free environment, and mainly focuses on the routing performance by fully making use of the available location. The potential excessive and uncontrolled usage of location information raises severe concerns of privacy in mobile and pervasive environments [2].

The first thing worth paying attention to is the scale of privacy problem. In daily life, people may not see privacy implications in revealing their location except in special circumstances. You probably do not care if anyone discovers where you were at 10:30a.m. yesterday, but if all of your movements are recorded every 5 seconds with foot accuracy, you might start to see things differently. In addition, network communications make the observation, propagation and processing of information on-the-fly, and the memory of information can be potentially unlimited. The scale of this problem changes thoroughly.

Specifically in ad hoc networks, most of available geographic routing protocols [3, 8, 11, 19] require each node to periodically update its current location to its neighbors and possibly remote servers. No further control over the exposure of node location actually encourages the potential abuse. For example, tracking of individuals becomes possible. By analyzing a history of tracking records, personal sensitive information such as health condition, social interest, and political tendency, etc., are easily revealed. Moreover, location sniffers are freely able to exchange their observation data or sell them to any interested parties, such as ad companies, to make profits. It should not be surprised that some day you receive tremendous advertisements related to your interests that even you are not aware of.

Although the problem is clear, preserving location in ad hoc networks, especially for geographic routing, appears to be quite challenging. The expected solution is not only required to prevent location sniffing from outside of the network, but also from the inside. "Good" nodes are not supposed to learn others' location because the lack of proper centralized administration in ad hoc networks enforces limited pressure of investigation and legal pursuits for information leaking. In centralized wireless networks, such as cellular networks, the problem of location exposure also exists, but typically users have a privacy agreement with their operators. User location is only collected by base stations rather than by other users. However, it is not so practical for each node to enforce privacy policies in ad hoc networks. Traditional privacy preserving approaches [2, 4] based on centralized control become not suitable in our context.

Traditional security solutions for content privacy such as IPSec [9] are not applicable in our context either since routing information is not within the scope of protection. Link level encryption of location enforces a severe computational burden to the network while it only prevents external eavesdroppers. Very little work on privacy issue in ad hoc networks has been done despite a considerable number of secure routing protocols [5, 13, 17, 20] has been proposed. The most related work is ANODR [10] proposed by Kong and Hong. ANODR achieves an anonymous on-demand routing and provides route untraceability by using route pseudonymity.

Since to restrict or stop using location information is not a solution to the problem, we expect to address the location privacy issue, while still exploiting location to achieve efficiency in routing functionality. Our solution tries to circumvent those difficulties by dissociating location information with identity. We believe that location information itself is not as sensitive as the simultaneous presence of the subject identity and its location. So, in this paper, we explore a way of breaking the linkability of location to subject identity for the the sake of location privacy preservation. We consider that the location privacy problem must be addressed at multiple network layers to avoid single penetration leading to compromising the whole. Our work focuses on the routing design, especially the geographic routing since it poses the major concern of the problem. And, we assume that upper layers sufficiently take care of the privacy issue. The organization of this paper is as follows: Section 2 presents the general model of geographic routing we consider, and discusses privacy threats based on this model. Section 3 discusses our proposed scheme and describes three components towards the anonymity of geographic routing, which is followed by security analysis in Section 4 and performance evaluation in Section 5.

2 Geographic Routing Model & Threats

Geographic routing utilizes location information to improve routing, which eliminates some limitations of topology-based routing. The forwarding decision at each node depends only on the local information, and becomes nearly “stateless”. The frequently used strategy to make such forwarding decision is greedy forwarding, where the forwarding node will forward packets to the closest neighbor to the destination. The packet will ultimately reach the destination using the same strategy iteratively.

Typically, each node is assumed to be able to determine its own location, such as by aid of a GPS receiver. By exchanging location locally, nodes are also able to obtain their neighbors' location. However, obtaining the destination's location relies on a so called location service. In the literature, several location service schemes [1, 11, 19] are available to help a node locate its intended destination.

We summarize the geographic routing model considered in our work as follows:

- Local location update (LLU) - each node periodically updates its current location to neighbors along with its identity, which enables the building up of a neighbor table at each node.
- Remote location update (RLU) - each node periodically updates its current location with identity to remote location servers according to a specific location service algorithm, and reactively responds to location requests (the location server typically responds on behalf of it.)
- Location request (LREQ) - a source node who does not have the location of the intended destination initiates a LREQ message to the corresponding server obtained by the specific location service algorithm. An LREQ message attaches the location and identity of the source so that the response of requested location could reach the original requester.
- Data delivery - during the data forwarding process, each packet attaches the location and identity of the destination so that geographic forwarding strategies could be applied and the intended destination could receive its data.

Based on the geographic routing model, we can see that, from a malicious party point of view, a node can keep on collecting the interested party's location through following ways:

1) Observe the interested node's location if it happens to be inside the radio range, or to be a relay/eavesdropper over the path of the node's update, request, and data packets.
2) Keep on making itself a location server of the interested node by following the location service algorithm applied, and accepting its remote location update.
3) Keep on initiating location requests to the interested node's location servers.
4) Collect the related information from its colluding nodes. Information from different sources could be merged.

As we can see, the location and identity is a basic doublet for distributing throughout the network so as to support the functionality of geographic routing. In the meanwhile, it is also the explicit source of threats to location privacy. The uncontrolled location exposure along with identity makes it a severe concern of privacy as we discussed earlier. Malicious tracking and analysis of sensitive personal information are made easy.

To preserve the basic privacy rights, we propose a privacy-aware geographic routing scheme by decoupling the location and its subject identity. Furthermore, we attempt to maintain the comparable performance with the original geographic routing.

3 The Proposed Scheme

The basic idea in our scheme is that location with identity is much more valuable than location itself to identify the subject and its related personal information. Thus, the unlinkability of location to its identity could achieve a certain level of location privacy in the sense that the adversary with only location information cannot derive the identity of the subject who is at the location. In fact, certain location may have indication of identity of the subject, when it's strongly associated with a subject, e.g. the location of the dean's office. However, we will only consider the general case in this paper, and we assume that location itself is not enough to derive its subject.

The proposed scheme consists of three main components: anonymous neighbor table (ANT), anonymous greedy forwarding (AGFW), and anonymous location service (ALS).

3.1 Anonymous Neighbor Table (ANT)

Neighboring exchange is the main way of disseminating location in geographic routing. Nodes build their neighbor tables by exchanging their location and identities. Maintaining an anonymous neighbor table (ANT) presents one of the main challenges of the design. It is the essential component for supporting AGFW and ALS to be discussed in the next few sections.

3.1.1 The First Attempt

Every node in the network periodically broadcasts a *hello* message to indicate its existence and to update its latest position. The *hello* message is constructed like $\langle HELLO, n, loc, t_s \rangle$, where n is a pseudonym of the sender, loc is the current position, and t_s is a timestamp. n is randomly generated by the sender for each hello message, and the frequency of sending hello messages is typically based on the node mobility. A simple method to reduce the probability of n collisions in the neighborhood is by performing a hash over a locally generated pseudorandom number and its identity to get n, denoted as $n = hash(pr, id)$. The hash function could be any "collision-resistant" hash algorithm.

Based on periodic neighboring exchanges, each node builds up its ANT with an entry like $\langle n, loc, t_s, t_o \rangle$, where t_o is timeout for this entry. And routing decisions are made as discussed in the last section. It is notable that a snapshot of ANT at certain moment may have more than one entry for the same neighbor in this scheme, because the receiver of *hello* messages is not able to correlate two messages sent by the same neighbor, which is also a desirable feature we expect for anonymity. However, in order to take care of the forwarding decision involved with its old pseudonym, each sender of hello messages should memorize some old pseudonyms it generated. Due to the continuous timeout of table entries, it does not need to memorize too many but two latest ones. Thus, if it receives a packet intended for any of two latest pseudonyms, it should accept the packet.

However, multiple-entry for one neighbor may lead to ineffective forwarding decision. For example, the previous hop selects n_1 as the next relay just because n_1 is in best position, but it didn't notice that n_2, indicating a fresher position of the same neighbor as n_1, is in a better position. There is a basic solution to this issue. The original forwarding strategy has to go through a little modification. That is, not only the position but the freshness should also be considered in the forwarding decision. Forwarding could be better if the node movement is predictable, for example, velocity and direction are available with position. It's preferable to choose a fresher position rather than the best one in the ANT to improve the forwarding performance.

The first attempt of ANT builds an AGFW-supporting neighbor table, which does not require a node disclosing its true identity. However, authentication of neighboring nodes has not been considered yet. Potential spoofing attackers are not banned from the network as, for example, the attacker could forge a lot of hello messages with arbitrary pseudonyms to severely degrade the performance and to mislead the forwarding direction. Therefore, we require an authenticated ANT, where a node needs to be authenticated to other nodes but should not disclose its identity to any party including the communicating one. In general, providing security while maintaining privacy proves to be a great challenge. We will propose a solution based on Ring Signature.

3.1.2 Authenticated ANT Based on Ring Signature

The authenticated ANT is based on Ring Signature [16] to achieve $(k+1)$-anonymous neighbor table. We consider a neighbor table as $(k+1)$-anonymous, if and only if any neighbor in the table is indistinguishable from other k legitimate users. Therefore, the larger k is, the stronger anonymity we have.

A ring signature scheme provides *signer-ambiguity* in the sense that the verifier is not able to determine the identity of the actual signer among a set of signers with size r. This set of signers is called a ring. There are two typical operations of a ring signature scheme: *ring-sign* and *ring-verify*. *Ring-sign* takes the message to be signed, all public keys of members in the ring, and the private key of the actual signer as input to compute the final ring signature. The verifier of a ring signature could use *ring-verify* to check the validity of the signature but is not able to determine who actually generated it.

The basic operations of the scheme based on ring signature are illustrated in Algorithm 3.1.

Node A borrows public keys $\{KU_1, KU_2, ..., KU_k\}$ from certificates $\{cert_1, cert_2, ..., cert_k\}$ of signers $\{N_1, N_2, ..., N_k\}$ to *ring-sign* its *hello* message with its private key KR_A. And all involved public keys are attached to the message in the form of certificates for

Algorithm 3.1: AUTHENTICATED ANT(A, B)

```
comment: A simple example of AANT with node A and B
$A : m \leftarrow \langle HELLO, n, loc, ts \rangle$
$rsig_A \leftarrow \text{RING-SIGN}(m, KU_1, KU_2, ..., KU_k, KU_A, KR_A)$
$A \rightarrow * : \langle m, rsig_A, cert_A, cert_1, cert_2, ..., cert_k \rangle$
$B : \text{RING-VERIFY}(m, rsig_A, cert_A, cert_1, cert_2, ..., cert_k)$
```

the sake of verification at the recipient. Furthermore, to avoid correlation of two transmissions with the same set of signers, the sender should randomly select k public keys among all valid users. As one of A's neighbors, node B can be sure that the sender is an authorized user in $\{A, N_1, N_2, ..., N_k\}$ after signature verification, but it cannot determine who among them actually signed this message. By applying ring signature, ANT can achieve important security property of authentication as well as $(k+1)$-anonymity.

3.2 Anonymous Greedy Forwarding (AGFW)

AGFW achieves anonymous data delivery by avoiding the explicit specification of destination identity. It relies on an anonymously maintained neighbor table (ANT) to make routing decisions. AGFW does not attempt to guarantee the packet delivery since no recovery mode is considered when a topology dead-end happens. There are varieties of recovery mechanisms proposed in the literature [12]. We consider further study on a certain recovery mechanism could be possible.

The data packet header of AGFW is constructed as

$$\langle DATA, loc_d, n, trapdoor \rangle$$

where loc_d is the location of destination, n is the pseudonym of next hop relay, and *trapdoor* is a value that can only be opened by the intended destination. It is important for achieving destination anonymity. With a proper *trapdoor* included, the destination's identity could be avoided while its location has to be presented by the sender. By trying opening the *trapdoor*, a node could determine if it is the intended recepient. One possible way of achieving the expected trapdoor function is encrypting some data with the destination's public key,

$$trapdoor = KU_d(src, loc_s, tag_d)$$

where KU_d is the public key of the destination, loc_s is the location of the source and tag_d is some data like "Hey! You are the destination!", which lets the node know that it is the target if it could properly decrypt the message. Here we assume that each node has a valid certificate signed by a trusted third party like a certification authority (CA), and that the source is able to know the destination's certificate somehow, or it stores the certificate beforehand. Thus, the source has a reliable public key of the intended destination.

On receiving a data packet, the node first decides if it is the intended next relay node by checking n. If n is not the pseudonym of the node, it will simply discard the packet. Otherwise, it will continue the forwarding process. It finds out the closest location in the neighbor table towards the destination, and transmits the packet to the neighbor with the associated pseudonym.

By following the described forwarding process, a data packet goes towards loc_d anonymously without disclosing any identity of the source, destination, or relays. However, there are two questions remaining, (1) how the destination receives the packet, and (2) how forwarding process stops. In fact, a desirable feature here is that we do not require each forwarding node to waste computing resources on opening trapdoors for destination detection. The committed forwarder, who owns n, attempts on opening the *trapdoor* only when it enters the last hop region. A node determines the last hop region by checking whether loc_d is inside its radio range. If the node successfully opens the *trapdoor*, forwarding stops. If not, it continues to forward to a closer neighbor as discussed earlier. Once the node in the last hop region finds that it can neither open the *trapdoor*, nor have a closer neighbor towards loc_d than itself, it locally broadcasts the packet with n set to 0, which we call "the last forwarding attempt". A pseudonym equal to 0 indicates to all receivers that they should try opening the trapdoor, and no more forwarding is required. As we can see in the discussed forwarding process, the destination will be able to receive the data when it is the forwarder in the last hop region or is one of the receivers of the last forwarding attempt. Furthermore, forwarding will stop when the destination either accepts the data or is unreachable.

All packet transmissions are local broadcasts so that it will disclose neither sender's nor receiver's MAC address by specifying a predefined broadcast address. In fact, the sender in AGFW will only know the receiver's pseudonym as what we want to achieve in ANT discussed later. The sender never knows which MAC address to specify for the receiver.

However, it should not set its own MAC address either because of the potential linking to its location. We explain

```
Algorithm 3.2: OnReceivePacket($p\langle DATA, n, loc_d, trapdoor\rangle$)

procedure TryForward($p$)
  $N \leftarrow$ ChooseNextHop($ANT$)
  comment: ChooseNextHop returns a best suited pseudonym
  if $N \neq me$
    then ForwardTo($N, p$)
         return ( true )
    else return ( false )

main
  if $n \in \{pseudonyms\}$
    then if $loc_d \cdot loc_{me} \leq RadioRange$
           then if open($trapdoor$)
                  then Accept($p$)
                  else if TryForward($p$) = false
                         then LastHopAttempt()
           else if TryForward($p$) = false
           then Stop()
         comment: Forwarding stops, recovery mode could
                  be further considered
  else if $n = 0$
    then if open($trapdoor$)
           then Accept($p$)
           else Discard($p$)
  else Discard($p$)
```

how this linking is possible as follows. Since our protocol is not designed to be route untraceable, the eavesdropper can easily correlate the last hop to the next hop transmissions along the same route by checking if packets have the same *trapdoor* information. Thus, if a packet is detected from MAC address A, the overhearing node can find out from its record history the last hop packet on the same route and the pseudonym n_A included. Obviously, n_A can be confidently associated with A. Consequently, the location associated with n_A is identified.

Another issue is that typical ad hoc medium access control protocols such as IEEE 802.11 [6] do not provide broadcasting as reliably as unicast. To achieve reliable local transmission of packets, a network layer acknowledgment could be used. Once the current forwarding node receives the data, it initiates an acknowledgment for the packet. The ACK packet is also locally broadcasted for anonymity. It includes the information uniquely determining the packet received. Furthermore, to reduce message transmission, the ACK packet can be piggybacked on a data packet to be sent, and it does not necessarily acknowledge only one received packet at a time.

We summarize AGFW into some basic pseudocodes in Algorithm 3.2.

3.3 Anonymous Location Service (ALS)

Another important component for a complete geographic routing scheme is location service that we have not yet discussed so far. In case the source node does not know the location of the destination, it should be able to retrieve it through a location service. However, to achieve an anonymous location service is very challenging as well. In this section, we propose a scheme based on DLM, a scalable location service proposed by Xue et al. [19].

In DLM, the network is divided into grids of the same size. Each node could determine some special grids, where its location servers are, by mapping its identity to it. Thus, node identity and a certain set of special grids have established a fixed association of location service, which is publicly known. Each node will periodically update its location to its associated grids. And, any querying node could initiate a location request to an associated grid of the requested node to retrieve the location.

DLM provides an efficient way of managing location service, but without anonymity considered. The updater and requester will have to expose their location and identities to the location server in DLM. Our proposed anonymous location service attempts to dissociate a node's location and its identity but does not provide updater anonymity in terms of hiding its identity. The basic idea is that the updater will encrypt its location and its identity before sending it to the location server, and the po-

Algorithm 3.3: ANONYMOUS LOCATION SERVICE(A, B, S)

```
comment: An example of ALS
$A \rightarrow S : \langle RLU, ssa(A), E_{K_B}(A, B), E_{K_B}(A, loc_A, ts)\rangle$
$S$ : STORE$(E_{K_B}(A, B), E_{K_B}(A, loc_A, ts))$
$B \rightarrow S : \langle LREQ, ssa(A), E_{K_B}(A, B), loc_B\rangle$
$S \rightarrow B : \langle LREP, loc_B, E_{K_B}(A, loc_A, ts)\rangle$
```

tential location requester can retrieve it and decrypt it to obtain the location. The whole process of location updating and querying will not simultaneously expose the location and identity of any of the three parties. A basic instance of our scheme is illustrated in Algorithm 3.3. For simplicity, only three nodes A, B and A's location server S are involved in this instance. Before we discuss how the scheme works, we introduce basic notations used in our illustration. $ssa(x)$ is the application of a server selection algorithm over value x. If it is applied on a node's identity, ssa returns the target grid where the node's location server is. We denote an encryption operation over message m with public key K as $E_K(m)$.

As we can see in Algorithm 3.3, node A uses its real identity to determine the remote proxy but encrypts its location with B's public key. Thus, any node other than A and B will not be able to read A's location including the intended location server, though they do know where it is stored. Furthermore, A includes another component in the update message, $E_{K_B}(A, B)$, which is very important for the anonymity of querying nodes, is used as an index for the location server to store updates. On receiving the request from B, S is able to respond with the requested location by finding the entry with $E_{K_B}(A, B)$. Meanwhile, B does not need to worry about the exposure of its identity to sniffers, relays and the location server S.

Our proposed scheme avoids the explicit exposure of both location and identity in a traditional location service. However, the updating node has to identify all its possible senders and has to update the location server accordingly. Otherwise, some nodes may not be able to reach it. We consider this as a limitation of the proposed scheme. However, in practice, a node may not need to hide its identity or location all the time. Possibly a heterogeneous location update scheme could be much optimized. Once the node does not need a strict privacy protection any more, it can switch to a normal location service in order to reduce the effort needed to be accessed by potential senders.

Another problem is that the requester might have a potential exposure risk. Since the index part $E_{K_B}(A, B)$ is a fixed block of data, a sophisticated attacker may find a matching identity with a certain probability by collecting enough certificates or computing it exhaustively. An alternative scheme is that the requester does not provide the component $E_{K_B}(A, B)$, but the location server will return a set of encrypted locations that might be intended for different possible senders. However, as a trade of anonymity, the communication and computation overhead increase.

4 Security Analysis

The three proposed components AGFW, ANT, and ALS are expected to collaboratively realize anonymous geographic routing with no compromise of its functionality and performance. But they are not designed to be route-untraceable. The path that a packet follows could be roughly estimated from the cleartext of locations in the packet since geographic routing basically follows a physically shortest path. The objective of this work is to dissociate location information with identity by anonymizing communications where location information has to be in cleartext. In AGFW, what a sniffer can observe is that packets are going towards certain locations. But it cannot determine who is sending to whom. Thus, location privacy could be protected in the sense that no node exposes its identity and location simultaneously.

The authenticated ANT achieves a $(k + 1)$-anonymous neighbor table to support AGFW. The node sending *hello* messages can be authenticated by the verifier, but cannot be identified among a set of ambiguous signers. From the performance perspective, the scheme has to make a tradeoff between the anonymity requirement and communication overhead in terms of number of bytes to be transmitted. The larger the set of ambiguous signers is used, the stronger the anonymity the sender has, but with more certificates to transmit. To reduce the explicit communication overhead due to certificate attachments, a sender may only specify identities or serial numbers of those certificates, and allow explicit request for required certificates in case the verifier does not have them. The number of explicit requests are expected to decline significantly after the network boots up for a period. Furthermore, in the scheme we assume that a node has enough valid certificates beforehand for ring signature use as it must have a certificate of the node it is going to communicate. In fact, it's another important topic of security for key management, which is out of the scope. Our basic assumption in this work is that a legitimate node has its valid certificate obtained from an external certification authority. In addition, the node might need to retrieve enough of them for ring signature scheme before entering the network.

Achieving a scalable location service is one of the main challenges in the geographic routing research community. But in our context, it is even more challenging to achieve an anonymous location service that coordinates three parties (updater, server, and requester), while protecting their privacy. In the proposed ALS, an updater does not attempt to hide its identity but instead has its location encrypted. The requester will not need to expose its identity to the location server for retrieving location. Furthermore, location servers are also protected from being identified. As we discussed, the main limitation of the scheme is that the updater is supposed to anticipate its potential senders in order to update location servers properly. However, in a realistic application, nodes can apply heterogeneous location update strategies to adapt its variable anonymity requirements.

5 Performance Evaluation

We implemented AGFW with the first version of ANT to examine performance of the new protocol behaviors. In the simulation, we did not incorporate ALS so as to focus our evaluation on the major routing part. Since ALS does not essentially change the message exchange of the protocol, the performance is expected to be similar to the original location service. With extra message bits and limited cryptographic operations involved, one might also expect it to elegantly degrade a bit. The performance of ring signature based ANT varies with the specific requirements of authentication and anonymity. Potentially more byte-cost could be expected.

The metrics we used for the performance evaluation mainly include:

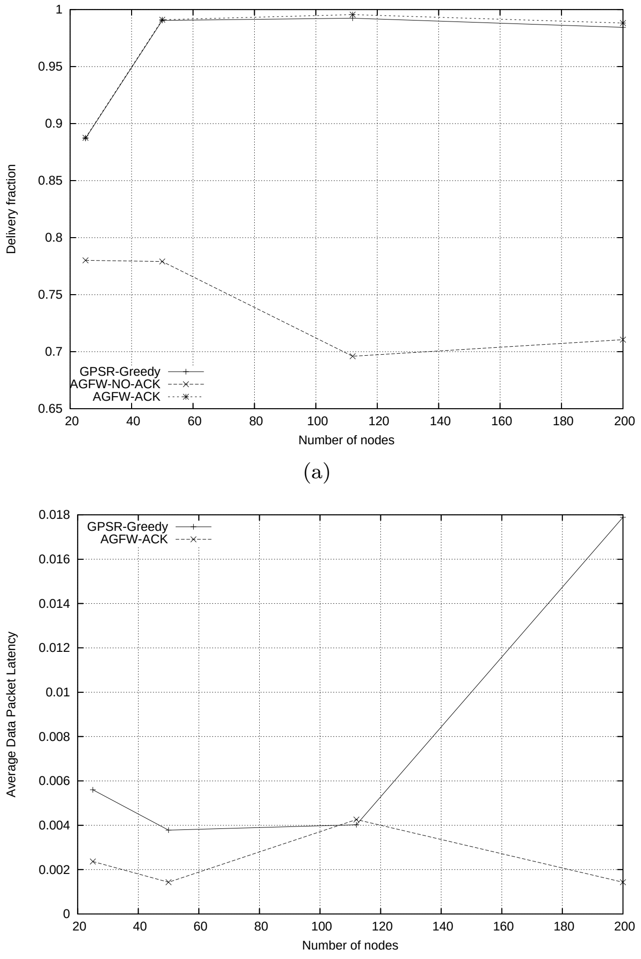
1) *Packet delivery fraction* - the fraction of the data packets sent out by sources that are delivered to destinations.

2) *End-to-end packet latency* - the average delay for a packet to be delivered from the source to the destination.

5.1 Simulation Model

We use NS-2 with CMU wireless extensions, which is wided accepted as the simulation environment for network research. Our implementation is based on the original codebase of GPSR [8]. The distributed coordination (DCF) of IEEE 802.11 [6] is used as the MAC protocol in our simulations. Typically, a unicast transmission in IEEE 802.11 uses Request-To-Send (RTS) and Clear-To-Send (CTS) control packets as virtual carrier sensing for reducing the well-known hidden terminal problem. Furthermore, each data transmission is followed by an ACK. However, a typical broadcast packet only uses CSMA/CA.

AGFW requires data packets being sent by local broadcasts in order to provide sender and receiver anonymity.



(a)

(b)

Figure 1: (a)End-to-end packet delivery fraction (b)End-to-end data packet latency

Thus, without virtual carrier sensing, potentially more packets may get collided due to the hidden terminal problem. As we mentioned in Section 3, a network layer acknowledgment could help. In this case, the local broadcast employed in our scheme is equivalent to a unicast, except that 802.11 uses typical MAC addresses while AGFW uses identity pseudonyms for unlinkability.

In our cryptographic implementation, the size of pseudonym (i.e. n) is equal to that of a typical MAC address. Thus, we do not think that pseudonym applied in the protocol is an extra requirement for packet size. For *trapdoor*, we mentioned that public key encryption can be applied. The underlying idea is that we do not assume extra key exchanges involved in our scheme, provided that public-key cryptography support is a prerequisite. However, we suggest a lower cost symmetric encryption if a proper key exchange scheme is in place. In our simulations, the size of *trapdoor* does not exceed 64-byte since it is obtained from the RSA [15] encryption with a 512-bit public key. A typical public-key encryption needs 0.5ms while the decryption needs 8.5ms for a portable computer processor [17]. Our simulations include a proper processing delay for where it applies.

Each simulation lasts for 900 seconds of simulation time. There are 50 nodes uniformly distributed in a network area with dimension 1500 × 300. Each node has a nominal radio range $250m$, and can move up to $20m/s$ with a pause time $60s$ whenever it changes its direction. We simulate 30 CBR traffic flows originated by 20 sending nodes.

5.2 Simulation Results

Figure 1(a) shows the packet delivery fractions of three schemes with the increase of the network density. We implemented a simple form of AGFW with no packet acknowledgment for comparison. Obviously, the delivery fraction is not satisfactory due to numerous packet collisions without ACKs and retransmissions. And it gets worse when more nodes entering the network lead to potentially more contentions and hidden terminals. AGFW with ACK capability has almost same performance as the original GPSR-Greedy. It indicates that the change of protocol behaviors for anonymity requirements in AGFW does not affect its routing reachability much.

Figure 1(b) shows the comparison of average end-to-end latency of data packets. An interesting result we obtained is that the packet latency of both schemes does not make much difference when the network has a modest node density, i.e. when the number of nodes is no larger than 112 in our simulations. Intuitively, AGFW should have a longer latency due to its larger packet size and the cryptographic processing delay. In fact, GPSR-Greedy does not get more advantages here: (1) The design of AGFW avoids the cryptographic processing overhead being spreaded over all intermediate nodes. Only those nodes within the range of the destination are required to try opening *trapdoor*. Therefore, a very limited number of nodes are affected by the processing overhead. (2) As we mentioned earlier, a typical 802.11 unicast involves a virtual carrier sensing by exchanging RTS and CTS, which contributes to the major part of the long latency of packet delivery in GPSR-Greedy. AGFW does not make handshakes before transmitting packets, where it could save a bit of waiting time, though the saved time is still limited. The reason is that we may also expect more packet collisions in AGFW due to potentially more hidden terminals with no RTS/CTS enabled. As a result, the time saved in skipping RTS/CTS is partially spent in potentially retransmitting packets and waiting for ACKs.

All positive and negative contributions to the latency lead to a limited difference of both schemes in terms of end-to-end packet latency. However, as indicated in the figure, when the network density becomes high, GPSR-Greedy presents a significant increase of packet latency due to relatively more failures of making handshakes and hence the time wasted on backing off and retries.

6 Conclusions

In this paper, we addressed a very critical issue in geographic routing protocols, i.e. how to guarantee location privacy protection while location information is used to maintain the efficiency of geographic routing. We circumvent traditional methods like spacial and temporal cloaking to design an anonymous geographic routing for preserving location privacy. We propose three components to achieve anonymity of communication as well as to maintain the functionality of a typical geographic routing protocol. Note that the forwarding strategy we used in this paper is greedy forwarding, because usually greedy forwarding has a satisfactory delivery performance even in a modest-density network. To avoid a simple dead end when local maximum happens, recovery strategies like perimeter forwarding [8] could be applied. We consider that it should not be difficult to extend the scheme to incorporate extra recovery mechanisms based on our approach. It will be our future work to extend the scheme.

References

[1] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A distance routing effect algorithm for mobility (dream)," in *Proceedings of Fourth Annual ACM/IEEE International Conference in Mobile Computing and Networking (MobiCom)*, pp. 76–84, 1998.

[2] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46–55, 2003.

[3] P. Bose, P. Morin, I. Stojmenovi&cacute, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *Wireless Networks*, vol. 7, no. 6, pp. 609–616, Nov. 2001.

[4] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *The First International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pp. 31–42, 2002.

[5] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: a secure on-demand routing protocol for Ad Hoc networks," in *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking* (MobiCom 2002), pp. 12-23, Sept. 2002.

[6] IEEE Computer Society, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Standard 802.11, 1999.

[7] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, Kluwer Academic Publishers, vol. 353, pp. 153–181, 1996.

[8] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Mobile Computing and Networking*, pp. 243–254, 2000.

[9] S. Kent and R. Atkinson, *Security Architecture for the Internet Protocol*, RFC 2401, 1998.

[10] J. Kong and X. Hong, "Anodr: Anonymous on demand routing with untraceable routes for mobile ad-hoc networks," `citeseer.ist.psu.edu/kong03anodr.html`

[11] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris, "A scalable location service for geographic ad-hoc routing," in *Proceedings of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom'00)*, pp. 120–130, 2000.

[12] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad-hoc networks," *IEEE Network*, vol. 15, no. 6, pp. 30–39, 2001.

[13] P. Papadimitratos and Z. J. Haas, "Secure routing for mobile Ad Hoc networks," in *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, pp. 27–31, 2002.

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

[15] R. L. Rivest, A. Shamir, and L. A. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

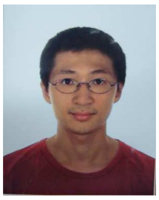
[16] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, pp. 552–565, Springer-Verlag, 2001.

[17] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A secure routing protocol for Ad Hoc networks," in *Proceedings of IEEE International Conference on Network Protocols(ICNP)*, pp. 78–89, Nov. 2002.

[18] I. Stojmenovic, "Position-based routing in ad hoc networks," *IEEE Communications Magazine*, vol. 40, no. 7, pp. 128–134, 2002.

[19] Y. Xue, B. Li, and K. Nahrstedt, "A scalable location management scheme in mobile ad-hoc networks," in *26th Annual IEEE Conference on Local Computer Networks (LCN'01)*, pp. 102–111, 2001.

[20] M. G. Zapata and N. Asokan, "Securing Ad Hoc routing protocol," in *Proceedings of the 2002 ACM Workshop on Wireless Security (WiSe 2002)*, pp. 1–10, Sept. 2002.



Zhi Zhou is currently a Ph.D. student of School of Computer Engineering in Nanyang Technological University, Singapore. He received his Bachelor's degree in Computer Science from Fudan University, China in 2002. His major areas of interest include wireless and ad hoc networks, routing security and location based services.

Kin Choong Yow received his Bachelor's degree in Electrical and Electronics Engineering with Honours from the National University of Singapore in 1993. He obtained his Ph.D. from the University of Cambridge, England in 1998. He is presently serving in the Nanyang Technological University (NTU), Singapore as an Assistant Professor and he is currently the Sub Dean of the School of Computer Engineering. He also leads the MANET group in the Centre for Multimedia and Network Technologies (CeMNeT) in NTU. His current research interest includes Multimedia communications, Wireless communication technologies, and Mobile Ad-hoc Networking.