Securing Geographical Routing in Mobile Ad-hoc Networks

Vivek Pathak
Department of Computer Science
Rutgers University
vpathak@cs.rutgers.edu

Danfeng Yao *
Department of Computer Science
Rutgers University
danfeng@cs.rutgers.edu

Liviu Iftode †
Department of Computer Science
Rutgers University
iftode@cs.rutgers.edu

July 9, 2008

Abstract

We present geographical secure path routing, an infrastructure free geographic routing protocol, that is resilient to disruptions caused by malicious or faulty nodes. Geographic locations of anonymous nodes are authenticated in order to provide location authentication and location privacy simultaneously. Our protocol also authenticates the routing paths taken by individual messages. This provides a basis for geographic security policies. This paper discusses the design and attack resistance of the secure geographic routing protocol. The overhead of location authentication is investigated under various scenarios through network simulation. Results show that the presence of malicious nodes increases the routing path length. A data delivery rate of better than 80% is sustained even if 40% of the nodes are malicious.

1 Introduction

Ad-hoc networks are becoming increasingly important with greater availability and popularity of networking enabled devices. These infrastructure free networks, which are assembled on the fly, are vulnerable to malicious nodes and other adversaries. Both the routing mechanism and the routed data flow are potential targets. Geographic routing is an established protocol for routing in ad-hoc networks [1, 2, 3]. It relies on nodes knowing their geographic locations, and using their one hop neighbors for routing packets to target geographic destinations. Location awareness simplifies ad-hoc routing but also raises privacy concerns. Adversarial nodes may collaborate to track node locations thereby violating location privacy. Securing geographic routing while simultaneously protecting location privacy is the focus of this work.

Securing ad-hoc routing is challenging because of the lack of pre-existing routing and security infrastructures. Nodes must create the routing infrastructure without using global knowledge. Lack of secure node identification is an additional challenge. Malicious or compromised nodes may pose as new nodes, or as known good nodes. Existing secure ad-hoc routing proposals have assumed various degrees of infrastructural support for addressing these challenges. In contrast, we use anonymous nodes and a cryptographic

*This work has been supported in part by the Rutgers University Computing Coordination Council (CCC) Pervasive Computing Initiative Grant.

†This work has been supported in part by the NSF grant CNS-0520123.

protocol for securing ad-hoc geographic routing in an infrastructure free manner. Our solution trades off security infrastructure for computational and messaging overhead. Ongoing improvements in the capabilities of networked devices make this choice reasonable.

We design an infrastructure-free secure geographic routing protocol. Our protocol protects location privacy and requires that nodes be able to determine their own geographic location. Geographic routing protocols are known to be particularly susceptible to location errors and attacks [4, 5]. Our protocol authenticates geographic locations thereby making it robust against malicious nodes. It also prevents malicious nodes from being used for routing. The routing paths taken by messages can be authenticated, which allows the development of new security policies, such as trusting message content if the message is routed through designated safe areas. In contrast to existing location authentication research, our approach does not require out-of-band communication or shared secret initialization.

1.1 Our solution

We propose geographical secure path routing (GSPR) for securing ad-hoc routing against malicious nodes and passive adversaries. The routing protocol operates on location aware anonymous nodes to provide privacy preserving secure geographic routing for ad-hoc networks. The protocol has the following goals:

- Route messages to desired geographic locations in the presence of malicious nodes. Detect and avoid bad geographic regions containing malicious or faulty nodes.
- Authenticate self-generated public keys and geographic locations of nodes on the routing path.

This paper describes the GSPR protocol in detail. Attacks from malicious nodes and passive adversaries are analyzed in order to demonstrate the correctness and attack resistance of the protocol. The overhead introduced by the protocol is modeled analytically and investigated in various operational and attack scenarios using the NS2 network simulator [6].

The rest of this paper is organized as follows. An overview of geographic routing is presented in Section 2. Section 3 discusses the geographic authentication model used for the proposed secure routing protocol. Section 4 describes the GSPR protocol. The attack resistance of GSPR is demonstrated in Section 5. The performance of GSPR in various operational and attack scenarios is investigated in Section 6. Related work is discussed in Section 7, and Section 8 presents the conclusions and future work.

2 Preliminaries

Geographic routing is a well researched approach for ad-hoc routing [1, 3, 7, 2]. Nodes are expected to know their own geographic locations, and to share it with their one-hop neighbors through periodic beacons. The periodic beacons allow each node to know the geographic locations of its one-hop neighbors.

Messages carry their target locations as they are routed though the ad-hoc network. Under the assumption of bidirectional connectivity, geographic routing can be efficiently implemented on a planar sub-graph of the one-hop connectivity graph. A number of planarization approaches have been developed, and each can

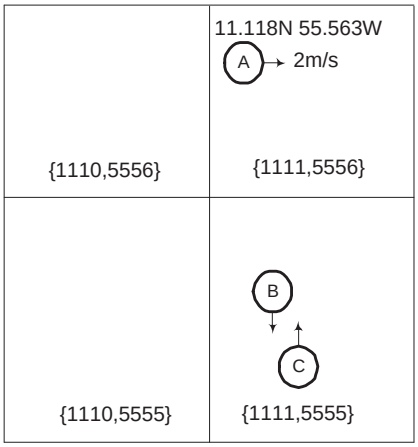

Figure 1: An example of geographic location and integer co-ordinates. Co-ordinates are shown in curly brackets, e.g., $\{1110, 5556\}$. 11.118N, 55.563W is a geographic location.

lead to a different subset of neighbors to use for routing. Once the choice of neighbors to use for routing is made, the next hop can be determined by greedily selecting the next hop to minimize the remaining distance to the target geographical location. Greedy forwarding fails if there is no next hop among the neighbors which is closer to the destination. In this case, geographic routing makes progress by entering the perimeter mode, in which the next-hop is selected to traverse the perimeter of the region where greedy forwarding fails. Perimeter mode forwarding continues as long as there is no better greedy next hop neighbor and the initial location where the perimeter mode started is not revisited again. Geographic routing is simple and efficient. The state required at each node depends only on the node density.

The underlying assumptions of bidirectional and distance based connectivity have been challenged through experimental studies in [7]. The same study also notes that these non-malicious errors are solvable through improved planarization techniques. Therefore we do not consider violations of bidirectional connectivity and transmission range in the rest of this paper. Prior research has also noted that geographic routing is susceptible to location errors and attacks [4, 5]. These security attacks are the basic focus of our geographical secure path routing protocol.

3 Model

This section discusses the assumptions, the problems being solved, and the notation used in the paper. The participants in our geographical secure path routing protocol are referred to as nodes. The nodes are located in an integer vector space as shown in Figure 1. The nodes may be stationary or mobile. These two cases are considered separately. The correctness and attack resilience of routing is first established for the stationary case. The effects of mobility on correctness, attack resilience, privacy, and performance are studied thereafter.

3.1 Assumptions

All the nodes are assumed to know their geographic locations. The protocol operates on integer co-ordinates which can be constructed by scaling the geographic co-ordinates with a global constant. For example, as shown in Figure 1, the integer co-ordinates $\{1111, 5556\}$ of node A can be derived from the geographic location 11.118N, 55.563W by applying a scaling factor of 100. The scaling factor is assumed to be a predefined constant well known to all the nodes. Nodes generate their own public-private key pairs. We also make standard assumptions about the non-invertibility of popular cryptographic functions.

Nodes are identified through short-lived temporary pseudonyms. These pseudonyms are also used as the physical level node identifiers in order to prevent physical level identification attacks. Pseudonyms are constructed from a pseudo-random number generator. This ensures that pseudonyms can not be used to derive real node identity. Nodes are assumed to silence transmissions while changing pseudonyms. Nodes can also estimate the number of one-hop neighbors either by listening to local transmissions or by knowing the node density. These capabilities are sufficient for using mix zones. Mix zones are regions with sufficient node density which can provide large enough anonymity sets to support the desired level of anonymity [8]. Changing pseudonyms within mix zones prevents node tracking by making it impossible to associate a series of pseudonyms with a physical node. This protects the location privacy of participating nodes.

We assume that promiscuous mode reception is enabled on the networking adapters. Nodes overhear all transmissions in their one-hop neighborhood. This enables detection of malicious activity in the one-hop neighborhood. The underlying data link layer is expected to handle the hidden terminal problem, packet collision, and asymmetrical connectivity issues. Node connectivity is assumed to be symmetrical and resilient to packet losses caused by collisions or jamming.

Node connectivity is assumed to have a range limited one-hop neighborhood. The one-hop neighborhood is bounded by a maximum distance R. Nodes farther apart than R are never one-hop neighbors. The maximum range distance is assumed to be a fixed global constant, which depends on the the link layer technology being used. For example, in an outdoor IEEE 802.11 peer-to-peer wireless network, the maximum range is limited to a few hundred meters. Violations of connectivity radius are detected, and result in the offending node being eliminated from the secure routing protocol. Limited connectivity radius is a reasonable assumption in the context of standardized wireless networking hardware.

3.2 Definitions

Consistent with traditional geographic routing [1], all the nodes are located on a plane. Every node p has a geographic location: $\text{Location}(p) \equiv (\mathbf{p_x}, \mathbf{p_y})$. The integer co-ordinates p_x and p_y of the node are computed by scaling the geographic location with the global scaling factor. Each node p has a public key K_p such that a message m can be secretly sent to p by encrypting with the well known public key as $K_p(m)$. The corresponding private key K_p^{-1} is only known to p, and can not be computed from K_p. All the messages transmitted from p are digitally signed with the private key K_p^{-1}. The notation used in this paper is summarized in Table 1.

Our GSPR protocol resists attacks from malicious and faulty nodes. Malicious nodes may intentionally try to give incorrect responses while faulty nodes may be attacked and have incorrect inputs to offer. Honest nodes are distinguished from malicious or faulty nodes as follows:

$f \circ g(x)$	Function composition of f and g, i.e. $f(g(x))$.
$f^{(k)}(x)$	The function f applied k times on x.
$\{x, y, z\}$	Message containing strings x, y and z.
K_p	Public key of the node p.
K_p^{-1}	Private key of the node p.
$K_p(x)$	A string x encrypted with the public key K_p of p.
$\{x\}_p$	A message consisting of string x signed by node p.
r_p	A pseudo-random number generated by node p.
$(\mathbf{p_x}, \mathbf{p_y})$	The geographic location of node p, also represented as Location(p).
(p_x, p_y)	Integer co-ordinates of node p derived by scaling its geographic location.
$\mathcal{N}(p)$	The set of nodes in the one-hop neighborhood of node p.

Table 1: Notation

DEFINITION 1 (Honest Node) *An honest node knows its correct geographical location, follows the maximum range constraint, and executes the GSPR routing protocol correctly. Otherwise, the node is called malicious or faulty.*

The ad-hoc network consists of honest and malicious nodes. These nodes may be placed at arbitrary geographical locations. Nodes become candidates for geographic routing depending on their geographic location. Routing paths consist of sequences of nodes. Each node on the routing path is responsible for forwarding the message towards the geographic destination. Given a node responsible for forwarding the message at a given hop, we define its honest witness node as follows:

DEFINITION 2 (Honest Witness) *An honest node is an honest witness for a forwarding node if it is in the one-hop neighborhood of the forwarding node, the previous-hop node, and the next-hop node for the given geographic routing path.*

The presence of honest witnesses allows the protocol to secure geographic routing while forwarding messages. We define an *honest witness network* as an ad-hoc network that has honest witnesses for all of its nodes.

DEFINITION 3 (Honest Witness Network) *A network is an honest witness network if there is an honest witness for every possible routing path of length* 3*.*

Geographic routing allows packets to be routed to destination locations. The routing protocol will return a failure message if there are no nodes in the one-hop neighborhood of the target location. If the one-hop neighborhood of the target location has one or more nodes, then each of them is considered a valid destination. Our GSPR protocol operates only on honest nodes. The routed packets are expected to reach target locations by using secure routing paths only.

DEFINITION 4 (Secure Routing Path) *A secure routing path consists of a sequence of honest nodes, each of which is the one-hop neighbor of its predecessor.*

Category	Specific Threat
Routing attack	Dropping messages
	Routing in the wrong direction
Data attack	Payload modification
	Control data modification
Malicious node	Reporting false location
	Directional transmission
	Transmission power changes
	Tracking node location

Table 2: Overview of the threats handled by GSPR

Participation in our protocol allows nodes to find a secure routing path to the destination. The source node requests for a node located near the desired geographic location. GSPR finds an honest node in the one-hop neighborhood of the location if possible. The returned response contains the public key of the discovered node and the secure routing path to it. Message integrity is guaranteed for both query and response messages. Next-hop nodes for the query messages are determined by geographic routing while the response message is source routed on the reverse path. The security properties of GSPR protocol are listed below:

DEFINITION 5 (Properties of GSPR protocol) *Given a message $\mathfrak{M}$ starting at node s with the destination geographic location D, a geographic secure path routing (GSPR) protocol is secure if the following properties hold in an honest witness network:*

- *If there is a secure routing path $\mathcal{S}(s,d)$ from the source node s to a node d located within one-hop distance of the destination location $\mathcal{D}$, then $\mathfrak{M}$ is routed to d.*
- *The destination node d receives the secure routing path $\mathcal{S}(s,d)$.*
- *On receiving the returned response, the source node s gets the correct public key K_d of the destination, and the secure routing path $\mathcal{S}(s,d)$ traversed by the message.*

3.3 Threat model

The routing protocol may be attacked for a number of reasons. Adversaries may create the false appearance of being at a location in order to gain additional privileges. For example, access to a classified document may be denied while away from a safe area, but approved within it. Adversaries may also have the simple motivation of rendering the routing mechanism unusable. These motivations are considered in this section. A summary of the threat model is also provided in Table 2.

A number of threats are expected in the wireless ad-hoc networking environment. Both the routing mechanism and the routed data may be attacked. Attacks can be mounted on the routing mechanism by dropping or incorrectly forwarding messages. Incorrect forwarding means forwarding messages towards incorrect directions, and includes classical attacks like routing loops and wormholes. These attacks against

geographic routing are considered in the security analysis. Attacks on routed data could target the payload or the control data required for protocol operation. Control data susceptible to modification includes node identifiers, node locations, and other data fields governing the routing protocol. The threat of data modification is also considered in our analysis.

Nodes are also vulnerable to attack in ad-hoc networking environment. Compromised nodes can be controlled by an attacker causing them to behave maliciously or incorrectly as defined in Definition 1. Malicious nodes may also collude to continuously track the location of a node thereby violating its location privacy. In case of nodes having home locations, location privacy violations also make anonymity violations more likely. Threats from compromised or malicious nodes are considered in our analysis along with possible attacks on location privacy and anonymity of nodes.

A number of low level attacks are possible against wireless ad-hoc networks. The jamming attack blocks radio transmissions in a given geographic region thereby preventing the routing protocol from using that area. Jamming can be tackled with spread spectrum techniques [9]. Other low level attacks include transmission power changes and directional transmissions. These attacks are detected by the protocol and the responsible nodes classified as malicious.

4 Geographical secure path routing

The details of our geographical secure path routing protocol are described in this section. Each of the building blocks: geographic hashes, periodic beacons, geographic routing, and malicious node detection are presented below.

4.1 Geographic hashes

We develop a novel method, called geographic hashes, for encoding relative geographic positions of two nodes. Our solution associates unforgeable transient geographic hashes to relative geographic positions as shown in Figure 2. Nodes maintain a set of integer tokens called geographic hashes, which associate a secret with a geographic location. The secret can not be determined without being in the vicinity of the location, but its knowledge can be verified remotely. Geographic hash values are transient (i.e., are short-lived and keep changing while a node moves). They are computed by repeatedly applying cryptographic one-way functions on a random nonce chosen by the node.

Geographic hash contains a chain of hashed values. It has the following interesting property. Given a geographic hash value corresponding to a time period i, it is infeasible for any polynomial-time adversary to compute the geographic hash values corresponding to any subsequent time period $j > i$. On the other hand, the reverse computation can be easily done, that is, given a geographic hash value corresponding to a time period i, it is easy to compute the geographic hash values corresponding to the preceding time period $j < i$. Conceptually, this property is the opposite of a regular one-way hash function, where computing the hash value from an input is easy while finding the input corresponding to a hash value is hard. We are able to achieve our property by having a node to (1) pre-compute a chain of hash values $(r[1], \ldots, r[n-1], r[n])$, where $r[n]$ is the hash value of $r[n-1]$ and so on, and (2) use the chain in reverse order $(r[n], \ldots, r[1])$, that is, use $r[n]$ first, then $r[n-1]$ second, etc. Geographic hashes are created through modular arithmetic. Consider

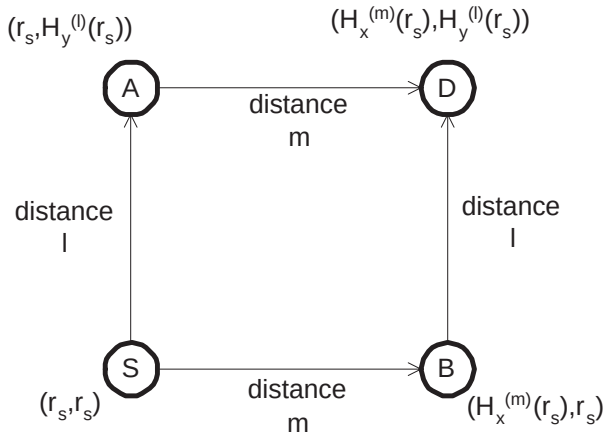

Figure 2: Using associative one-way functions to create geographic hashes. H_x and H_y are associative one-way functions which are applied to maintain geographic hash values at different geographical locations. r_s is a random nonce published by node s and serves as the geographic hash of s at s.

a large prime p and a generating number a, such that the function $f(x) \equiv a^x \bmod p$ maps $\mathbb{Z}_p^\star = \{1,\ldots,p-1\}$ onto itself. Each integer $h \in \mathbb{Z}_p^\star$ can be used to represent a one-way function $H(x) \equiv (a^h)^x \bmod p$ since the discrete logarithm problem of finding x, given $y = a^x \bmod p$, is believed to be NP-hard.. Formally, geographic hash is defined next.

DEFINITION 6 (Geographic Hash) *Each node A periodically publishes the following geographic hash parameters: a large prime p, a generator a for $\mathbb{Z}_p^\star = \{1,\ldots,p-1\}$, three integers $\zeta_A, \eta_A, \theta_A \in \mathbb{Z}_p^\star$, and a time interval Δ_A indicating the expiry time for a single version of the geographic hash.*

The geographic hash of A is initialized to (r_A, r_A) at A, where r_A is a random nonce selected by A. Successive versions $r_A[i]$ and $r_A[i+1]$ of the random nonce satisfy:

$$r_A[i+1]a^{\theta_A} \bmod p = r_A[i]$$

Each node B in the neighborhood of A computes the geographic hash of A at B as follows:

$$GH(A,B) = \left(r_A a^{\zeta_A \Delta x} \bmod p, r_A a^{\eta_A \Delta y} \bmod p\right)$$

where Δx and Δy are the differences in the integer co-ordinates related to the geographic location of A and B.

The geographic hash is a tuple of integers computed by repeated applications of one-way functions to the locally known geographic hash of a node. The functions are designed to encode the relative position across multiple applications on various nodes. Consider a component GH_y of the geographic hash after two translations Δy_1 and Δy_2 from A:

$$\begin{aligned} GH_y &= r_A.a^{\eta_A \Delta y_1}.a^{\eta_A \Delta y_2} \bmod p \\ &= r_A.a^{\eta_A \Delta y_1 + \eta_A \Delta y_2} \bmod p \\ &= r_A.a^{\eta_A (\Delta y_1 + \Delta y_2)} \bmod p \end{aligned}$$

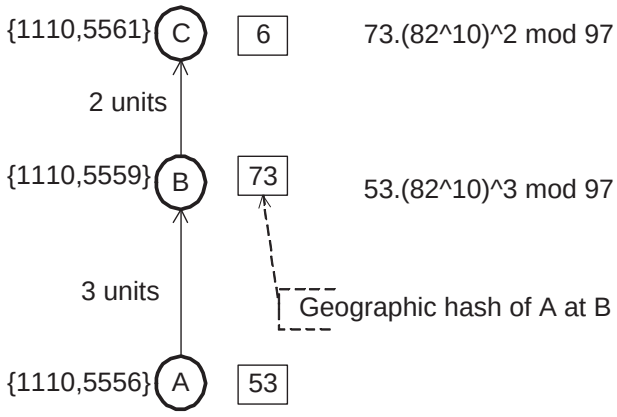

Figure 3: Example of geographic hash calculation. The node A selects $r_A = 53$ as its initial random nonce, 82 and 97 are the generating and prime numbers, and $\eta_A = 10$. The calculated y component of the geographic hash is shown in square boxes. An explanation of the value is given on the right.

The final value of the geographic hash component depends only on the initial random nonce r_A and the difference in integer co-ordinates. An example computation is shown in Figure 2. The y component of a geographic hash is being computed for the parameters $r_A = 53$, $\eta_A = 10$, prime number $p = 97$, and generating number $a = 82$. The final value of 6 encodes a relative translation of 5 units along the y axis because $6 = 53(82^{10})^5 \bmod 97$. Since the construction of geographic hashes encodes the relative geographic locations of nodes, the computed geographic hash values would be identical across different calculation paths.

4.2 Beacon

Nodes are required to know one-hop neighbor locations for geographic routing. This is achieved by having the nodes transmit a periodic beacon containing the node identifier and location. Nodes continuously listen for beacons from neighboring nodes. The information gathered from the received beacons is stored in memory in order to support geographic routing. Our protocol extends the beacon to include the public key and the random nonce selected by the node. This ensures that public keys of nodes are well known in the one-hop neighborhood. The beacon also includes locations and geographic hashes of neighboring nodes. This information helps in detecting malicious nodes as discussed in Section 5. Beacon messages are digitally signed with the private key of the node and are broadcast to all the one-hop neighbors.

DEFINITION 7 (Periodic Beacon) *Each node p periodically broadcasts a beacon message containing its geographic location, random nonce r_p, public key K_p, neighbor information list $\mathbb{Q}$, and a set of geographic hashes $\mathbb{G}$ known to it. The beacon message is broadcast to all nodes in the one-hop neighborhood $\mathcal{N}(p)$ of p as shown below:*

$$p \quad \rightarrow \quad \mathcal{N}(p) \qquad \left\{ \{p, Location(p)\}_p, r_p, K_p, \mathbb{Q}, \mathbb{G}, \mathbb{M} \right\}_p$$

where

$$\mathbb{Q} \equiv \left\{ \{q, Location(q)\}_q \mid q \in \mathcal{N}(p) \right\}$$

and

$$\mathbb{G} \equiv \left\{ \left\langle q, \zeta_q, \eta_q, \theta_q, \Delta_q, GH(q,p) \right\rangle \mid Distance(q,p) < 2R \right\}$$

and

$$\mathbb{M} \equiv \{ \langle q,\ Evidence \rangle \mid q \text{ is malicious} \}$$

The periodic beacon permits sharing the information which is used for validating routing actions. Sharing the neighbor information list $\mathbb{Q}$ helps in detecting false location attacks among the set of one-hop neighbors. The geographic hashes of nodes located within twice the maximum one-hop radius R are stored in memory. These geographic hashes are shared with neighboring nodes in order to detect malicious routing behavior beyond the one-hop neighbors.

4.3 Routing protocol

The GSPR protocol has a two step query-response messaging model. Payload and control data are sent towards the destination location as is done in traditional geographic routing protocols. The returning source routed acknowledgment completes the protocol. The routing protocol consists of the following operations: BEGIN FORWARDING, GEOGRAPHIC FORWARDING, LOCAL RESPONSE, RECURSIVE RESPONSE, END FORWARDING, REVERSE RESPONSE FORWARDING, and VERIFICATION. The additional stage of malicious node detection is treated separately in the next section because it encompasses all the operations described above. Malicious node detection results in nodes being detected as malicious or the neighborhood being classified as a bad neighborhood. Malicious nodes are not used for routing and honest nodes do not forward messages if they are located in bad neighborhoods.

Consider a data payload being routed from source node the s to a destination node d near the target geographic location. Let $\{p_0, \ldots, p_k\}$ be the intermediate nodes on the geographic routing path. The messages involved in the GSPR protocol are described below. All the messages include source and destination identifiers. These common fields are omitted below for brevity.

- BEGIN FORWARDING
  This operation executes at the source node s and begins routing the payload to the destination geographic location $\mathcal{D}$. The message contains a location list, $\mathbb{L}_s = [\text{Location}(s)]$, which is initialized to contain the source location. The message also contains the random nonce r_s selected by the source node. The message type "forward" is included in order to distinguish it from other messages in the protocol. The following message is sent to p_0, the next-hop node for geographic routing:

$$s \quad \rightarrow \quad p_0 \qquad \{\text{Forward}, \mathcal{D}, r_s, \mathbb{L}_s, \text{Payload}\}_s$$

- GEOGRAPHIC FORWARDING
  This operation executes at each node on the routing path. At node $p_i \in \{p_0, \ldots, p_k\}$, the next-hop p_{i+1}

is determined according to geographic routing towards the destination location $\mathcal{D}$. The operation is triggered on receiving a message with the type "forward" from the previous hop p_{i-1}. The operation results in further forwarding of the message if the current node is not the destination node. The forwarded message contains the location list $\mathbb{L}_i$ which is constructed by appending the location of the current node to the location list received from the previous node p_{i-1}, i.e. $\mathbb{L}_i = \mathbb{L}_{i-1} + [\text{Location}(p_i)]$. The node p_i transmits the following message to the next hop p_{i+1}.

$$p_i \quad \rightarrow \quad p_{i+1} \qquad \{\text{Forward}, \mathcal{D}, r_s, \mathbb{L}_i, \text{Payload}\}_i$$

The operation also causes a "local response" message to be returned to the previous hop node p_{i-1}. The local response contains public key, node identifier, and geographic hash (see Definition 6) information $\mathit{I}(p_{i+1})$ about the next-hop node:

$$\mathit{I}(p_{i+1}) \equiv \left\langle K_{p_{i+1}}, p_{i+1}, \text{Location}(p_{i+1}), \text{GH}(p_{i+1}, p) \right\rangle$$

$$p_{i-1} \quad \leftarrow \quad p_i \qquad \{\text{Local Response}, r_s, K_{p_i}^{-1}(r_s), \mathit{I}(p_{i+1})\}_i$$

The purpose of the local response message is to help verification of the protocol operation beyond the one-hop region. It permits the previous hop p_i to check the validity of the public key $K_{p_{i+1}}$. The presence of the geographic hash of the next hop node permits the current node to verify that the next to next-hop location is genuine and correct.

- END FORWARDING
  This operation executes at the destination node d. Repeated forwarding of "forward" messages routes the payload to the destination. The destination node does not forward a message of type "forward" but runs the receive operation instead. On verifying the integrity of the received message, the destination sends back the "recursive response" to the source. The recursive response contains the location list $\mathbb{L}_d$ traversed by the geographically routed message. This list is used to route the recursive response back to the source. The recursive response also contains a list of public keys of the nodes on the reverse routing path, $\mathbb{P}_d = [K_d]$, which is initialized to contain the public key of the destination. The challenge nonce r_s and its response $q_d = K_d^{-1}(r_s)$ are also sent back towards the source.

$$p_{k-1} \quad \leftarrow \quad d \qquad \{\text{Recursive Response}, \mathbb{L}_d, r_s, q_d, \mathbb{P}_d\}_d$$

- REVERSE RESPONSE FORWARDING
  This operation executes at all the nodes on the routing path. The operation is triggered on the node p_i on receiving a "recursive response" message from the node p_{i+1}. The public key of the current node is appended to the public key list received in the incoming message: $\mathbb{P}_i = \mathbb{P}_{i+1} + [K_{p_i}]$. The location list $\mathbb{L}_d$ is checked to ensure the current node is on the reverse route. If so, the following message is transmitted to the next node p_{i-1} on the reverse route:

$$p_{i-1} \quad \leftarrow \quad p_i \qquad \{\text{Recursive Response}, \mathbb{L}_d, r_s, K_{p_i}^{-1}(q_{i+1}), \mathbb{P}_i\}_i$$

- VERIFICATION
  This operation executes on the source node after it receives the recursive response message routed back to it. On receiving the recursive response, the source gets a signed copy of the routing path $\mathbb{L}_d$

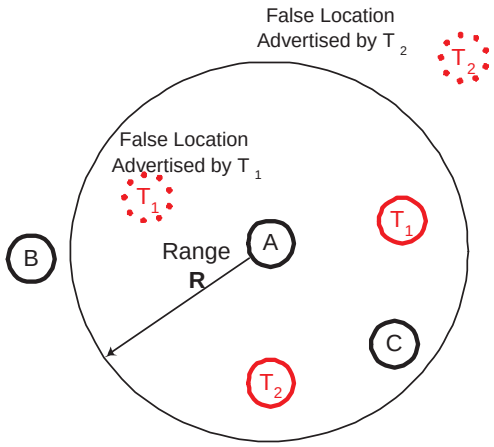

Figure 4: Different types of false location attacks

taken by its original forward message and the list of public keys $\mathbb{P}_0$ belonging to the nodes on the routing path. The source also gets the response $q_0 = K_{p_0}^{-1} \circ K_{p_1}^{-1} \circ \ldots \circ K_d^{-1}(r_s)$ to its challenge nonce r_s. The correctness can be verified using the public keys in $\mathbb{P}_0$ as:

$$r_s = K_{p_0} \circ K_{p_1} \circ \ldots \circ K_d(q_0)$$

This verification completes the protocol.

4.4 Malicious node detection

Malicious node detection is based on the broadcast nature of wireless communication and is modeled after the watchdog protocol [10]. Nodes listen to the transmissions of their neighbors in order to detect malicious nodes. Malicious nodes are not used for routing. Since the protocol operates in an honest witness network (see Definition 3), an honest witness node is always able to detect malicious activity and warn other neighboring nodes through its periodic beacon. Watchdog protocols are vulnerable to blacklisting attacks. Because the GSPR protocol uses temporary pseudonyms, a blacklisting attack can only have temporary effect. Avoiding temporary blacklisting is not a goal of the protocol, and is not considered further. Malicious nodes are detected both by checking for inconsistencies in the periodic beacons and by checking the correctness of geographically routed messages and their reverse source routed responses. Among the threats identified in Section 3.3, false location attacks are detected through inconsistencies in the periodic beacons. The remaining threats are handled by overhearing query and response messages in promiscuous mode. This permits detection of malicious forwarding.

4.4.1 Beacon validation

Periodic beacons are received from neighboring nodes. Recall from Definition 7 that beacon messages contain location information about one-hop neighbors and the geographic hashes of nodes within two-hops of the sending node. Beacons are stored in memory and their contents used for detecting malicious nodes.

Nodes launching false location attacks are detected by applying the range constraint R. Each node constructs a number of mappings from pseudonym to location, one received from each neighbor. Small inconsistencies in location are ignored as location errors. Larger inconsistencies permit the node to conclude that its one-hop neighborhood has malicious nodes. The precise threshold of location error limit depends on the GPS error rate. As shown in Figure 4, the node A can detect that the advertised location of T_2 is out of range. This node must be malicious and can be ignored for routing. A related situation occurs for T_1, which appears to be in the direction of B when it is not. The node A detects this malicious behavior with help of the node C. The beacon from C will contain either a different or an infeasible location for T_1. Thus A can determine that its one-hop neighborhood has malicious nodes.

The detailed procedure for detecting false location attacks based on received beacons is given in the algorithm below:

Algorithm 4.4.1 Detecting malicious nodes and false location attacks on node p.

```
1  Malicious := []
2  NeighborInfo := []
3  BadNeighborhood = False
4  while beacon := Recv_Beacon()
5        q := Sender(beacon)
6        if Distance(q, p) > R
7           Malicious := Malicious + {q}
8        for r ∈ Neighbors(q)
9           if Distance(q, r) > R
10             Malicious := Malicious + {r}
11          if Distance(p, r) < R ∧ r ∉ NeighborInfo
12             Malicious := Malicious + {r}
13          if ¬(NeighborInfo(r) ≃ Location(r, beacon))
14             BadNeighborhood = True
15       NeighborInfo[q] := beacon
```

The malicious node detection procedure maintains the list of malicious nodes and a flag indicating if it is located in a bad neighborhood. The state associated with malicious node detection is temporary and will be cleaned on expiry of the pseudonym. As shown in Algorithm 4.4.1, received beacons permit nodes to update their neighbor information and to decide if the sender is malicious. Malicious nodes are detected in case of infeasible distances between nodes as shown on lines 6, 9, and 11. The node can also decide the neighborhood is bad if there are large disagreements in neighbor node locations as shown on line 12. While nodes located in bad neighborhoods do not forward messages, they continue to transmit the beacon in order to propagate geographic hashes and their knowledge of malicious nodes in the one-hop neighborhood.

4.4.2 Forwarding validation

Operating in promiscuous mode permits overhearing wireless transmissions of one-hop neighbors. Incorrect forwarding is detected and the incorrectly forwarding nodes classified as malicious. Nodes forward signed

messages during GEOGRAPHIC FORWARDING and REVERSE RESPONSE FORWARDING stages. These operations can be abstracted to a simple multi-hop ad-hoc forwarding protocol as follows: Let *A*, *B*, and *C* be successive hops on a routing path. All the transmissions made by *B* must be received by *A* because of the one-hop neighbor relationship. Therefore, *A* can detect if *B* fails to forward a message by listening for the next-hop transmission. In this case the honest previous hop node *A* can identify the malicious node *B* without the need for other honest witness nodes. Similarly, *A* can check the overheard next-hop transmission (from *B* to *C*) for malicious payload modification or control data modification. Data tampering by *B* is detectable as the message from *B* to *C* is digitally signed with the private key of *B* which can be verified by *A* because of being in the one-hop neighborhood. The node *A* detects that the next-hop for its message is a malicious node.

In case both *B* and *C* are malicious, *B* can forward the message correctly to *C*, and later collude by not reporting a malicious forwarding by *C*. This collusion case is detected by the honest witness node *G* which is the one-hop neighbor of *A*, *B*, and *C*. Note that such a node must exist in an honest witness network (see Definition 3). The honest witness *G* will observe the malicious action of *C*. The honest witness will broadcast this malicious action to its one-hop neighborhood through the periodic beacon. Since *A* is in the one-hop neighborhood of *G*, it will also mark the node *C* as malicious. But *A* also overhears the forwarding of its message from *B* to *C*. Thus *A* detects that its message is being forwarded through a malicious node. On encountering malicious nodes, the previous hop node *A* will find another route to the destination or send back a routing failure to the source.

Nodes check the integrity of messages and feasibility of routing paths. Both GEOGRAPHIC FORWARDING and REVERSE RESPONSE FORWARDING stages require messages to carry the geographic location list $\mathbb{L}$ of nodes on the routing path. Forwarding nodes check the integrity of the digitally signed message and the feasibility of the routing path. Since the system has the global upper limit *R* on the range of nodes, a feasible path can not have successive nodes *A* and *B* at a distance greater than *R*:

$$R \geq \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2} \tag{1}$$

If this range constraint is violated, then the message is faulty or the previous hop node is malicious. Messages violating the range constraint are not forwarded and their previous-hop nodes are classified as malicious.

5 Analysis

An adversary can attempt to attack our routing protocol in a number of ways. As outlined in Section 3.3, the threats can be classified into two categories: attacks against the beacon and attacks against the routed messages. This section demonstrates how our protocol withstands the threats.

5.1 False location attacks

Beacons propagate neighbor information within the one-hop neighborhood. This information is used for geographic routing. Beacons are directly transmitted to one-hop neighbors. Transmission within the one-

hop neighborhood obviates the need for protecting against dropping messages and incorrect routing. Direct transmission also protects against payload and control data modification. Since beacons are digitally signed with the private key of the transmitting node, another node can not construct a false beacon. Malicious nodes can use beacons to launched false position attacks and try to cause incorrect routing. The following two lemmas show that these attacks are either detected or are harmless.

LEMMA 1 *Given a pair of honest nodes A and B within one-hop distance of each other, false location attacks which change the expected one-hop relationship to A or B are detected.*

PROOF: Let C be a malicious node with real location l_C. C launches a false location attack by reporting a false location $l_C^{'} \neq l_C$. Now consider the one-hop relationship of possible locations to A and B. Let L_A be the set of locations in the one-hop neighborhood of A, and L_B the set of locations in the one-hop neighborhood of B. The reported false location $l_C^{'}$ can change the expected one-hop relationship to A or B in one of the following ways:

- If $l_C^{'} \notin L_A \cup L_B$ and $l_C \in L_A \cup L_B$: The beacon of C is heard by at least one of A or B. The node hearing the beacon marks C as malicious because of the violation of range constraint as shown on line 6 of Algorithm 4.4.1. The other node learns about C being malicious through the periodic beacon of its peer (see the beacon specification in Definition 7). Thus both A and B detect the false location attack and mark C malicious.
- If $l_C^{'} \in L_A - L_B$, then A should hear the beacon and B should not. If $l_C \notin L_A - L_B$, then either A will not hear C's beacon or B will hear it. Since A and B are one-hop neighbors, they share the location information received from beacons. Both A and B will recognize that either A does not hear C's beacon or B does. This happens on lines 9 and 11 of Algorithm 4.4.1. In this case l_C is false because the range constraint is violated for one of two nodes. Beacon exchanges allow both A and B to detect the false location attack and mark C as a malicious node.
- The case when $l_C^{'} \in L_B - L_A$ follows by symmetry from the previous case.
- If $l_C^{'} \in L_A \cap L_B$ then both A and B should hear the node of the beacon. If $l_C \notin L_A \cap L_B$. then one of A or B will not hear the beacon directly but receive it indirectly through the other node. It will detect the false location attack at line 11 of Algorithm 4.4.1 and mark C as malicious. Beacon exchanges allow both A and B to share this information and mark C as a malicious node.

In each of the possible cases when the false location has a different one-hop relationship with A and B as compared to the real location, the false location attack is detected. □

LEMMA 2 *Given a pair of honest nodes A and B within one-hop distance of each other, false location attacks which do not change the expected one-hop relationship to A or B, do not affect the correctness of routing through nodes A or B in an honest witness network.*

PROOF: Let C be a malicious node at location l_C, and let C report a false location $l_C^{'} \neq l_C$. Let L_A be the set of locations in the one-hop neighborhood of A, and L_B the set of locations in the one-hop neighborhood of B. The reported false location $l_C^{'}$ has the same one-hop relation to A and B as l_C. The following cases can be distinguished by the one-hop relationship of C to A and B:

- If $l_C^{'}, l_C \notin L_A \cup L_B$, then C is not used for routing because of being out of range. Thus, routing through A or B is not affected.
- If $l_C^{'}, l_C \in L_A - L_B$, then A is in the one-hop neighborhood of C and hears all the transmissions from C. Note that C can only lie on routing paths through A because it is out of the range of B. If the reported false location $l_C^{'} \in L_A - L_B$ is not on a routing path through A, then it can not impact correctness of routing. If $l_C^{'}$ is on a routing path through A and X is the next-hop node on it, then X can either be within one-hop of l_C or not. If X is within one-hop of l_C, then the location attack does not affect routing because the next hop receives the message.

  If the next-hop X is not within one-hop of l_C, then by the honest witness assumption, an honest node G must exist within one hop of A, l_C, and X. G will observe the message from C, but not the forwarded message from the next-hop X. The witness G will broadcast the error to its one-hop neighborhood which includes A. Now A will either select a different next-hop than C or mark the neighborhood bad and return an error upstream to the previous hop.
- The case when $l_C^{'}, l_C \in L_B - L_A$ follows by symmetry from the previous case.
- The case when $l_C^{'}, l_C \in L_A \cap L_B$ follows by considering the arguments for the previous two cases.

In each of the cases either the message will reach the destination or an error notification will reach the source. Thus the location error does not affect the correctness of routing. □

5.2 Routing and message integrity

Preserving message integrity and correct geographic forwarding are required for secure routing. As shown in Lemma 1, beacon based malicious node detection detects large location errors which could affect routing. Undetectable location errors do not affect geographic routing as shown in Lemma 2. This leaves message dropping, routing in the wrong direction, payload modification, and control data modification as the remaining threats to consider. The following two lemmas show that message integrity is preserved, and that messages are routed to the correct destination location.

LEMMA 3 *Message integrity is preserved by the geographical secure path routing protocol operating in an honest witness network.*

PROOF: Consider the message $\mathfrak{M}$ starting at node s with destination location $\mathcal{D}$. Let the sequence of nodes $\mathcal{S}(s,d) \equiv \{s,\ldots,d\}$ denote a feasible geographic routing path from the source node s to a destination node d in the one-hop neighborhood of $\mathcal{D}$. Consider the forwarding of $\mathfrak{M}$ on three successive nodes A, B, and C on the routing path $\mathcal{S}(s,d)$. Let B be the node launching the attack and modifying either the payload or control data.

Recall the routing protocol described in Section 4.3. The GEOGRAPHIC FORWARDING operation on node B forwards the message to the next hop C. Since A and B must be one-hop neighbors by virtue of being successive hops on a routing path, A must overhear the transmission of B to C. If B modifies the contents of the message, A will detect the attack. Then A will mark the node B as malicious. The node A can now select another next-hop neighbor, or send back a routing failure if no suitable neighbor exists. In either case, message integrity is not violated. □

LEMMA 4 *In an honest witness network, geographical secure path routing protocol routes the messages to the correct next hop node.*

PROOF: Consider the forwarding of $\mathfrak{M}$ on three successive nodes A, B, and C on the routing path $\mathcal{S}(s,d)$. Let B be the node launching the routing attack. As outlined in Section 3.3, the following attacks are possible:

- *Dropping messages*:
  Recall the routing protocol described in Section 4.3. The GEOGRAPHIC FORWARDING operation executing on node B forwards the message to the next hop C. Since A and B must be one-hop neighbors by virtue of being successive hops on a routing path, A must overhear the transmission of B to C. If B drops the message, A will detect the missing transmission and mark B as malicious. The node A either selects another next-hop node or sends a routing error back to the source.

- *Routing in the wrong direction*:
  The GEOGRAPHIC FORWARDING operation is expected to route the message to the correct next hop node C. The malicious node B can launch one of the following attacks on the choice of node C:

  – The node C is not the correct choice among the neighbors $\mathcal{N}(B)$ of node B. Let $D \in \mathcal{N}(B)$ be the correct next-hop neighbor. By assumption of honest witness network, there must be an honest witness node G which is in the one-hop neighborhood of A, B and D. Since the message content can not be modified by Lemma 3, D will overhear the malicious forwarding by B and detect that $\mathfrak{M}$ should have been routed to it, but has been routed to C instead. D will mark B as malicious and share the information with G. G will send the error back to A.

  – Node B manufactures a false location for C. An honest witness G to A, B, and C will pass back the correct geographic hash of C to A through the beacon. On receiving a "Local Response" with incorrect geographic hash for the next hop C (see Section 4.3), A will detect it is in a bad neighborhood.

  In either case A selects another next-hop node or sends a routing error back to the source.

- *Collusion among malicious nodes*:
  If both the nodes B and C are malicious, and let C forward the message in an incorrect direction. The node B can collude with C and not report the error back to A. However, in the honest witness network, there must be an honest witness node G which is in the one-hop neighbor of A, B, and C. The node G will detect C dropping the message or routing in the incorrect direction. The proof follows from the previous two cases of this lemma.

In all the cases above the next-hop is selected correctly if it exists. If no appropriate honest node can be chosen, a routing error is sent back to the source. □

5.3 Security of geographical secure path routing

The security goals of geographical secure path routing follow from the resistance to false location attacks, correctness of routing, and preservation of message integrity. These properties have been established in the previous lemmas for honest witness networks.

THEOREM 1 *In an honest witness network, our geographical secure path routing protocol provides the following properties:*

- *If there is a secure routing path $\mathcal{S}(s,d)$ from the source node s to a node d located within one-hop distance of the destination location $\mathcal{D}$, then $\mathfrak{M}$ is routed to d.*
- *The destination node d receives the secure routing path $\mathcal{S}(s,d)$.*
- *On receiving the returned response, the source node s gets the correct public key K_d of the destination, and the secure routing path $\mathcal{S}(s,d)$ traversed by the message.*

PROOF: Consider the message $\mathfrak{M}$ being routed on $\mathcal{S}(s,d)$. At each hop the routing path malicious actions of neighboring nodes are either detected or are harmless for geographic forwarding from Lemmas 1 and 1. Hence the routing path avoids malicious nodes. Lemma 4 ensures that messages are routed to the correct next-hop nodes for geographic routing. Since geographic routing will consider all honest candidates for routing at every hop, the routing protocol will find a secure routing path if it exists. Message integrity is will be preserved at every hop according to Lemma 3. Thus the message $\mathfrak{M}$ will be delivered to d if there is a secure routing path $\mathcal{S}(s,d)$ from source to destination.

The destination node learns about the secure routing path because the "Forward" messages in the routing protocol (see Section 4.3) carry the list of node locations. Because each hop preserves message integrity, the received location list must be correct.

The source node receives the "Recursive Response" message by source routing on the reverse path. The message carries the public keys and locations of the nodes on $\mathcal{S}(s,d)$. Since message integrity is guaranteed at every hop, the received public key and routing path are correct. □

5.4 Node density requirement

The existence of honest witnesses is required to guarantee the security of geographical secure path routing protocol. Increasing node density makes it more likely that honest witnesses shall exist for every segment of a routing path. The practical way of determining node density would require a separate information channel because a single physical node can pose as multiple nodes. Nodes are therefore expected to have a side channel which can estimate node density in their one-hop neighborhood.

If a given fraction of nodes can always be assumed to be honest, then honest witness density can be related to node density. Honest witness node requirement can then be met by modifying the forwarding procedure, and requiring a higher node density.

LEMMA 5 *Let R be the node connectivity radius, such that nodes closer than R are always one-hop neighbors. If successive hops are selected within $\frac{R}{3}$ distance of each other, and if the honest node density is greater than $\frac{18}{R^2}$, then the network is an honest witness network.*

PROOF: Consider a square with diagonal $\frac{R}{3}$. Given that the honest node density is greater than $\frac{18}{R^2}$, each of the four quadrants of the square must have an honest node. By design each of the nodes in the four quadrants are one-hop neighbors of each others. Consider a message that enters the square at node A placed in one of

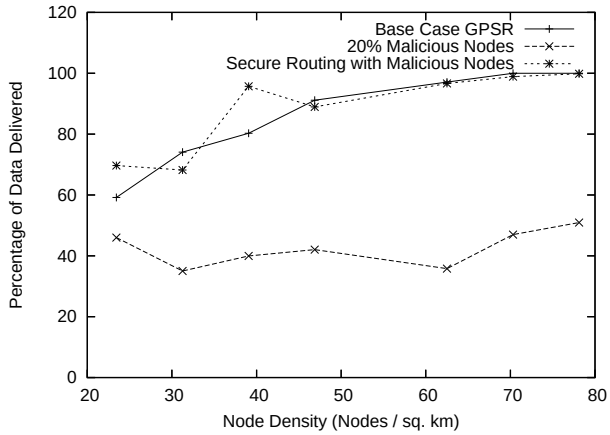

Figure 5: Data delivery vs. node density on a square area with stationary nodes.

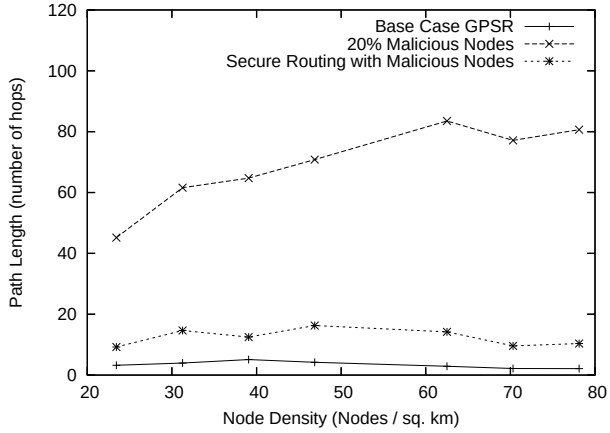

Figure 6: Routing path length vs. node density on a square area with stationary nodes.

the quadrants. One of the remaining three nodes must be the next hop because geographic routing routes by direction and the next hop must be within $\frac{R}{3}$ distance.

Without loss of generality, suppose the opposite diagonal node B is the next hop. The distance between A and B is at most $\frac{R}{3}$. Let C and D be the remaining two nodes in the square, and let X be the next hop from B. The distance between X and B is at most $\frac{R}{3}$ because successive hops must be selected within $\frac{R}{3}$ distance of each other. The distance between B and C is at most $\frac{R}{3}$ by construction. By triangle inequality, the distance between X and C is at most $\frac{2R}{3}$. Since A, B, and C are one-hop neighbors of each other, C is the honest witness for the routing segment A, B, X. If instead of node B, nodes C or D were chose as the next hop from A, the same argument would apply with B being the honest witness.

The plane can be tiled into squares of diagonal $\frac{R}{3}$, each having 4 honest nodes in order to have the honest node density of $\frac{18}{R^2}$. Therefore the above argument must apply at all locations. Because every routing segment has an honest witness node, the network is an honest witness network. □

Node density estimation is also required for privacy protection. Changing node pseudonyms in mix zones is sufficient for privacy protection [8]. Mix zones are regions with sufficient node density which can provide large enough anonymity sets to support the desired level of anonymity. The pseudonym changing procedure requires a sufficient number of nodes to silence their transmissions while changing pseudonyms. This prevents an attacker from associating two versions of the pseudonym to the same physical node thereby providing location privacy to the physical node.

5.5 Overheads

The overhead imposed by the geographical secure path routing protocol can be expressed in terms of the routing and deployment parameters. Consider a geographic routing path consisting of p nodes. Recall from Section 4.3 that the "Forward" messages contain the list of node locations. Then the message size overhead on the forward path is $\mathbf{O}(p)$. The computational cost on the forward path is two public key operations per forwarded message per node. One operation is for digital signature verification on the incoming message, and the second is for signing the outgoing message with the private key. The overhead on the return path is of the same order as that on the forward path.

Secure geographic path routing also introduces an overhead on the periodic beacon transmitted by the node. The beacon carries constant size overhead caused by the nonce, digital signature, and the public key (see Definition 7). It also contains neighbor location list, and the geographic hash list which linearly scale with the node density. Thus given a node density of ρ, the beacon overhead is $\mathbf{O}(\rho)$.

6 Performance analysis

Performance of geographic secure path routing is analyzed by the NS2 network simulator [6]. Although NS2 has support for wireless and mobile ad-hoc network simulation, geographic routing is not available in the standard NS2 code. Therefore, we use the patch provided by Kiess [11, 12] that maintains Karp's original implementation of GPSR [1]. The patch simulates IEEE 802.11 MAC layer with a node range of 500m. It provides support for mobility through the random way point model [13].

Geographic routing takes all routing decisions based on the local one hop neighborhood. Since secure routing detects and avoids malicious nodes, the changes to routing performance can be evaluated by changing the routing behavior to avoid malicious nodes. The remaining protocol operations just authenticate node locations and public keys without affecting routing. Therefore, we change only the local routing behavior of geographic routing in order to assess the performance of GSPR. We modified the simulator code to keep track of malicious or honest nodes. The routing mechanism was also modified to operate in the base mode or to avoid malicious nodes for routing.

The objective of this evaluation is to compare the routing performance and attack resilience of traditional insecure GPSR protocol against our proposed geographical secure path routing protocol. We select percentage rate of data delivery and the routing path length as the indicators of routing performance. The comparative evaluation of the two routing protocols is done for various combinations of node density, mobility, and the presence of malicious nodes.

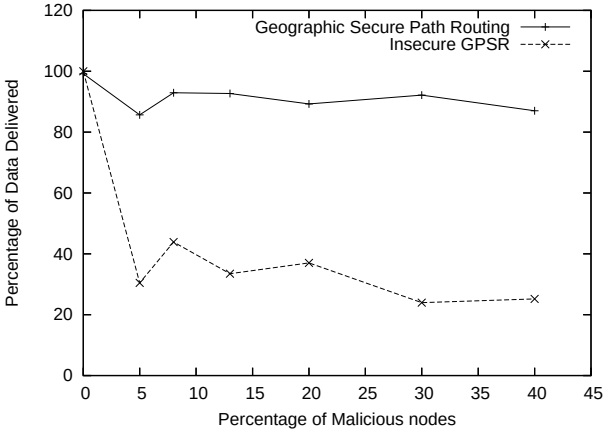

Figure 7: Effect of malicious nodes on data delivery.

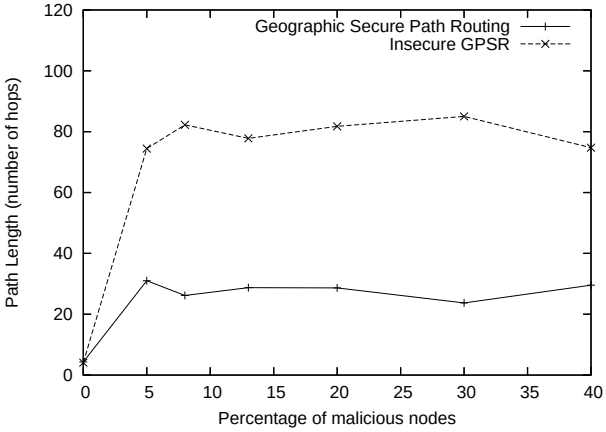

Figure 8: Effect of malicious nodes on path length.

6.1 Node density

GPSR and secure routing were simulated on NS2 using one constant bit rate source per node. Simulation is done with stationary nodes randomly placed on a square area with side 800m. The number of nodes was varied from 20 to 50 to create a number of node density scenarios. The simulation was run for 300 seconds of simulated time. An average of 10 runs was used in the following observations. The delivery rate was calculated by counting the number of application packets sent and received. Path length was computed by tracing GPSR packets through forwardings. As shown in Figure 5, the baseline GPSR protocol attains a high percentage of packet delivery. This is consistent with Karp's observations in [1]. It can be observed that introducing 20% malicious nodes severely impacts the effectiveness of GPSR as demonstrated by the reduction in delivery rate. Geographic secure path routing is resilient to the malicious nodes. Its delivery rate closely replicates the rate achieved by GPSR in a benign environment.

The impact of incorrect routing introduced by malicious nodes is shown in Figure 6. Given the node

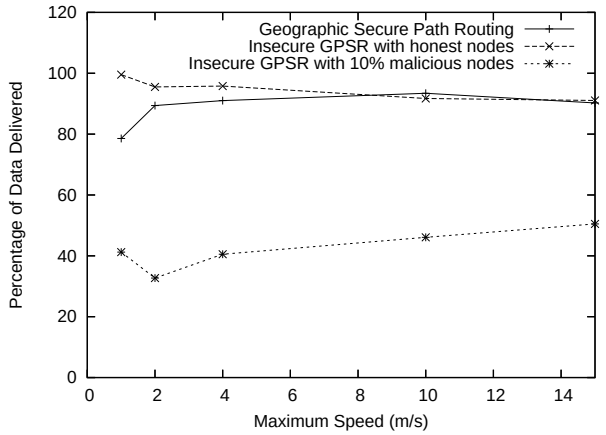

Figure 9: Effect of mobility on data delivery for baseline and with 10% malicious nodes.

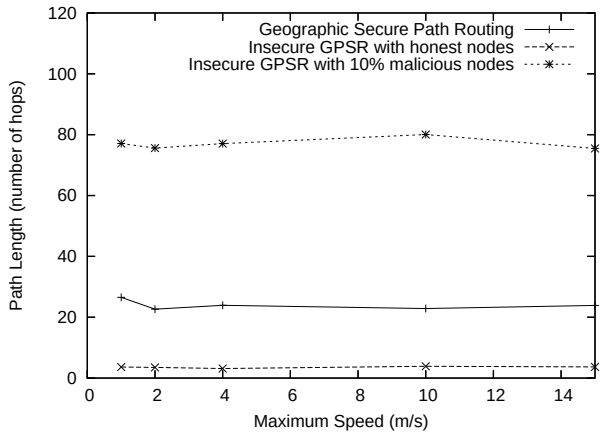

Figure 10: Effect of mobility on forwarding path length for baseline and with 10% malicious nodes.

range of 500m, and the $0.64km^2$ node placement area, we expect nodes to be about 2 hops away in the greedy routing case. This is consistent with our average reading of 3.38 hops for GPSR in benign environment. We also note that secure routing incurs an overhead on the routing path length by routing packets around malicious nodes. The average number of hops for geographic secure path routing with 20% malicious nodes is 10.37, an increase of about three times over GPSR in benign environment.

6.2 Effect of malicious nodes

Malicious nodes publish an incorrect location on their beacons. We modify the published location of malicious node to appear as the closest neighbor for the packet being forwarded. This allows malicious nodes to falsely become the next hop neighbors. Malicious nodes also forward the packet to a random neighbor in violation of geographic routing rules. The effect of malicious nodes on GPSR and geographic secure path routing is studied by running an NS2 simulation with varying proportions of malicious nodes on a

universe of 42 stationary nodes. These nodes are placed randomly in a rectangular area of 1.5km by 0.5km. Observations are collected by averaging the collected over 10 runs of 300 seconds simulated time each.

The data delivery achieved in presence of malicious nodes is shown in Figure 7. It can be observed that insecure geographic routing is very sensitive to malicious nodes. The delivery rate falls rapidly even with a small percentage of malicious nodes. The simulation also indicates that malicious nodes do not affect the delivery rate of our secure routing. The secure delivery rate falls from close to 100% in a benign environment to about 90% when 40% of the nodes are malicious. The effect on forwarding path length is shown in Figure 8. The path length shown by insecure geographic routing grows by orders of magnitude as the malicious nodes force the insecure protocol to route in incorrect directions. Geographic secure path routing incurs a more modest overhead by rejecting malicious nodes for routing.

6.3 Effects of mobility

The effect of mobility on data delivery rate is given in Figure 9. Mobility improves the data delivery rate of secure routing because mobility allows nodes to discover new honest nodes. This effect is also found in the insecure geographic routing to a smaller degree. Geographic routing in insecure environment becomes less effective with increasing mobility because of increasing chance of inaccuracy in one hop node locations. We also note that mobility reduces the routing path length of secure routing as shown in Figure 10. This happens because mobility increases the chance of discovering new honest nodes in the one hop neighborhood.

7 Related work

Secure routing in ad-hoc networks has been investigated by a number of prior works. Hu and Perrig propose the Ariadne protocol for securing on-demand and source routing protocols in ad-hoc networks [14]. Similarly, the SEAD protocol [15] secures distance vector routing in ad-hoc networks. Both the protocols requires a secure cryptographic initialization phase, but use highly efficient symmetric key cryptography. While our approach uses expensive modular arithmetic, it does not require secure initialization. A public key infrastructure based approach to securing distance vector routing is presented in [16, 17]. Public key infrastructure is also used in [18] to secure on-demand ad-hoc routing protocols. Threshold cryptography based secure ad-hoc routing is presented in [19]. All of these approaches are infrastructure oriented. Our secure routing protocol, in contrast, is infrastructure free.

The privacy threat posed by location aware devices has also been a topic of intense research. Defective or compromised devices could allow tracking of their users. A coalition of malicious nodes could co-operate to continuously track the geographic locations of correct devices (and their users). Existing research has taken two approaches for protecting user privacy: the first is to fudge the locations of identifiable nodes as in [20, 21]. The second is to use pseudonyms for temporary identification of nodes as proposed in [22, 23, 24, 25, 8]. We use the latter approach because secure geographic routing requires authenticated locations. Using temporary pseudonyms for location aware nodes allows us to provide location privacy and location authentication simultaneously.

A multi-hop anonymous challenge mechanism has been used by Mahajan et. al. [26] for detection of free riders in ad-hoc wireless network. Their mechanism requires two-hop transmission of challenge messages.

Our recursive challenge response is related to their mechanism in the sense that remote nodes are used for security processing. Unlike in our work, the authors do not study the effects of mobility.

A scheme based on secure cryptographic initialization is presented by Liu et. al. [27] to provide robust location estimation for sensor nodes in a hostile environment. Our approach is distinguished in the sense that we do not require any secure initialization. However, our approach is more expensive, requiring asymmetric cryptography. We discuss this as a future work in Section 8.

A statistical approach for robust localization using triangulation is described by Li et. al. [28]. Our approach is similar in the sense we allow the existence of malicious nodes. The difference in the approach is that we use cryptographic primitives at a higher level, and do not depend on physical level data. Other approaches include Sastry et. al. [29] who use echo latency for localization. We do not use out of the band communication for localization.

8 Conclusion and future work

We design and evaluate geographical secure path routing, a privacy preserving ad-hoc routing protocol, which geographically routes messages through anonymous nodes to destination locations. The secure routing protocol also authenticates the public key and the geographic location of destination nodes. This enables private communication with nodes located in a given geographic area.

Geographical secure path routing protocol requires associative cryptographic one-way hash functions for security. These hash functions are derived from the discrete logarithm problem which uses expensive modular arithmetic. This makes our protocol unsuitable for power limited devices. This limitation could be overcome by implementing the protocol with light weight cryptographic primitives.

Geographical secure path routing protocol was evaluated with the NS2 network simulator for various values of node density, node mobility, and the proportion of malicious nodes. Evaluation results show that the protocol tolerates malicious nodes with an increased routing path length. The geographical secure path routing protocol is also able to maintain a low loss rate even when the majority of nodes are malicious.

References

[1] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, August 2000, pp. 243–254.

[2] P. Bose, P. Morin, I. Stojmenovic;, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *Wirel. Netw.*, vol. 7, no. 6, pp. 609–616, 2001.

[3] G. G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," Information Sciences Institute, Research Report 180, March 1987.

[4] T. Leinmüller and E. Schoch, "Greedy routing in highway scenarios: The impact of position faking nodes," in *Third International Workshop on Intelligent Transportation (WIT)*, March 2006.

[5] S. Kwon and N. B. Shroff, "Geographic routing in the presence of location errors," *Comput. Networks*, vol. 50, no. 15, pp. 2902–2917, October 2006.

[6] "The Network Simulator ns-2 (v2.1b8a)," http://www.isi.edu/nsnam/ns/, October 2001.

[7] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic routing made practical," in *Proceedings of 2nd Symposium on Networked Systems Design and Implementation*. USENIX, 2005, pp. 217–230.

[8] S. Capkun, J.-P. Hubaux, and M. Jakobsson, "Secure and privacy-preserving communication in hybrid ad hoc networks," Ecole Polytechnique Fédérale de Lausanne, I&C Research Report 200444, May 2004.

[9] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: attack and defense strategies." *IEEE Network*, vol. 20, no. 3, pp. 41–47, May-June 2006.

[10] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *MOBICOM*, 2000, pp. 255–265.

[11] W. Kiess, H. Füßler, J. Widmer, and M. Mauve, "Hierarchical Location Service for Mobile Ad-Hoc Networks," *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, vol. 8, no. 4, pp. 47–58, October 2004.

[12] W. Kiess, "Hierarchical Location Service for Mobile Ad-Hoc Networks," Master's thesis, Department of Mathematics and Computer Science, University of Mannheim, 2003.

[13] C. Bettstetter, "Mobility modeling in wireless networks: categorization, smooth movement, and border effects," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 3, pp. 55–66, 2001.

[14] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks." *Wireless Networks*, vol. 11, no. 1-2, pp. 21–38, 2005.

[15] Y.-C. Hu, D. B. Johnson, and A. Perrig, "SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks." *Ad Hoc Networks*, vol. 1, no. 1, pp. 175–192, 2003.

[16] T. Wan, E. Kranakis, and P. C. van Oorschot, "Securing the Destination-Sequenced Distance Vector Routing Protocol (S-DSDV)." in *ICICS*, ser. Lecture Notes in Computer Science, J. Lopez, S. Qing, and E. Okamoto, Eds., vol. 3269. Springer, 2004, pp. 358–374.

[17] M. G. Zapata and N. Asokan, "Securing ad-hoc routing protocols," in *WiSe '02: Proceedings of the 2002 ACM workshop on Wireless security*, 2002.

[18] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An On-Demand Secure Routing Protocol Resilient to Byzantine Failures," in *WiSe '02: Proceedings of the 2002 ACM workshop on Wireless security*, 2002.

[19] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, no. 6, pp. 24–30, 1999.

[20] J. Lundberg, "Routing security in ad hoc networks," Helsinki University of Technology, Tech. Rep. Tik110. 501, 2000.

[21] A. Grlach, W. W. Terpstra, and A. Heinemann, "Survey on Location Privacy in Pervasive Computing," in *Proceedings of The First Workshop on Security and Privacy at the Conference on Pervasive Computing (SPPC)*, April 2004.

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

[23] X. Wu and B. K. Bhargava, "Ao2p: Ad hoc on-demand position-based private routing protocol." *IEEE Trans. Mob. Comput.*, vol. 4, no. 4, pp. 335–348, 2005.

[24] S. M. M. Rahman, M. Mambo, A. Inomata, and E. Okamoto, "An anonymous on-demand position-based routing in mobile ad hoc networks." in *SAINT*. IEEE Computer Society, 2006, pp. 300–306.

[25] B. Zhu, Z. Wan, M. S. Kankanhalli, F. Bao, and R. H. Deng, "Anonymous secure routing in mobile ad-hoc networks." in *LCN*. IEEE Computer Society, 2004, pp. 102–108.

[26] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Sustaining cooperation in multi-hop wireless networks," in *Proceedings of 2nd Symposium on Networked Systems Design and Implementation*. USENIX, 2005, pp. 231–244.

[27] D. Liu, P. Ning, and W. K. Du, "Attack-resistant location estimation in sensor networks," in *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*. Piscataway, NJ, USA: IEEE Press, 2005.

[28] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*. Piscataway, NJ, USA: IEEE Press, 2005, p. 12.

[29] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims." in *Workshop on Wireless Security*, W. D. Maughan and A. Perrig, Eds. ACM, 2003, pp. 1–10.