

Securing Geographic Routing in Wireless Sensor Networks

K.D. Kang, K. Liu, and N. Abu-Ghazaleh
Department of Computer Science
State University of New York at Binghamton
{kang, kliu, nael}@cs.binghamton.edu

Abstract

We consider the security of geographic routing (GR) that is widely used in ad hoc and wireless sensor networks due to its scalability. In GR, a node greedily forwards a packet to the neighbor that is closest to the destination. Thus, GR only requires a node to maintain the location information of its one hop neighbors. However, very little work has been done to secure GR. In a potential attack, malicious nodes may falsify their location information. Also, a malicious node can send an excessive number of packets to overload the receiving nodes and block legitimate packets from other sources. Alternatively, it can drop or misdirect received packets. To shed light on these problems, we propose an approach for robust GR via rate control, packet scheduling, and trust-based multi-path routing. In a simulation study, we also show that our robust GR can circumvent and route against attacks.

1 Introduction

Small, battery-powered, wireless sensors can be easily deployed for sensing in a number of important applications, e.g., habitat monitoring, disaster recovery, and battle field monitoring, without requiring any communication infrastructure. Secure routing in WSNs (Wireless Sensor Networks) is critical but challenging, because the sensors have very limited energy, bandwidth, and computational resources. In addition, they are often deployed in an open environment where physical security is unavailable.

In this paper, we consider the security of geographic routing (GR) in which a node greedily forwards a packet to the neighbor closest to the destination. GR is an attractive approach for routing in WSNs due to its low overhead and localized interactions. In GR, nodes only need to interact with their one-hop neighbors to exchange the location information and make localized forwarding decisions. Although GR is widely used in ad hoc and wireless sensor networks due to its scalability, very little work has been done

to secure GR.

Specifically, we aim to improve the robustness of the GR against two classes of attacks. In the first category of attacks, called flooding attacks, a malicious or faulty node can send an excessive number of packets to overload the receiving nodes and block packets originating from other sources. Alternatively, it can drop or misdirect received packets. This class of attacks are called blackhole and selective forwarding attacks [11]. Without any oversight and control mechanisms, GR may fail in the presence of these attacks. To shed light on the problem, we developed a trusted-based multipath routing protocol [1] that is resilient to blackhole/selective forwarding attacks. In this paper, we further extend it by considering rate control and packet scheduling to support resilience to flooding attacks. In our approach, an individual sensor node computes the expected data incoming rate from its neighbors based on the queries usually disseminated by the base station in WSNs. If a neighbor violates the computed bound, the receiving node drops excessive packets. The receiving node assigns low priority to the remaining packets from the suspicious neighbors (sending an excessive number of packets) to favor packets received from well-behaving neighbors. In addition, it forwards each of the remaining packets from the suspicious nodes to a single, randomly chosen neighbor rather than strictly following the trust-based multipath routing [1] or geographic routing protocol [12]. In this way, we can save precious resources such as energy to service more packets received from well-behaving nodes. By sending suspicious packets to randomly chosen neighbors, the node can also build the trust information of its neighbors that will not be selected in the original geographic routing such as [12] or our extended protocol [1] that only considers the blackhole/selective forwarding attacks. More details of our approach to defending against flooding and blackhole/selective forwarding attacks are discussed in Section 4.

The remainder of the paper is organized as follows. In section 2, background information about GR, localization, and location verification is given. Our threat model is described in Section 3. A robust GR protocol is proposed in

Section 4. Moreover, the security of the protocol and the related tradeoffs are discussed. In Section 5, we simulate part of our GR protocol and compare its performance to a well-known insecure geographic routing protocol [12]. The related work is discussed in Section 6. Finally, Section 7 concludes the paper and discusses possible future extensions.

2 Background

This section presents background information about GR, localization, and location verification needed to discuss the security issues of GR.

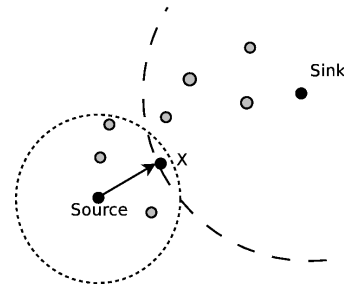
2.1 Geographic Routing

Geographic routing has two parts: geographic forwarding and face routing [12]. Geographic forwarding is a greedy routing algorithm based on geography. For a given node, all its one-hop neighbors closer to the sink belong to the *forwarding set* (FS). As shown in Figure 1(a), the node forwards an incoming data packet to the neighbor in the FS that is closest to the sink. GR is attractive, since it only requires nodes to maintain the location information of their one-hop neighbors. Also, routing decisions can be made locally and dynamically. However, geographic routing does not always succeed in the greedy phase. When the forwarding node, e.g., node x in Figure 1(b), has no one-hop neighbor closer to the sink than itself, it cannot further forward the incoming packet. Thus, the packet is stuck in a local minimum, called a *void*, where the FS is empty. In such a case, typically a complementary mechanism, e.g., face routing [12] or backtracking towards a beacon [5], is used to route around the void.

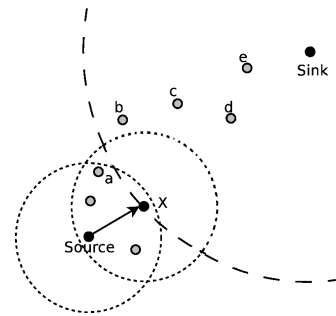
2.2 Localization

Usually, sensor nodes are not equipped with their own GPS (Global Positioning System) devices for several reasons such as the cost and size. Localization algorithms, in which sensor nodes do not require their own GPS devices, can be classified as follows.

- **Triangulation:** In this approach, the location of a sensor node is determined using trigonometry, i.e., lateration or angulation. Anchor nodes equipped with a positioning device such as a GPS are assumed to know their locations. Anchor nodes periodically broadcast their location information to all their one-hop neighbors. Lateration [16] is the calculation of position information based on the distance estimated from the anchors. Three distance measurements are required to localize a 2D position. In Figure 2, for example, a node



(a) X is the neighbor closest to the sink.



(b) Void: X is a local minimum.

Figure 1. Geographic Routing Example

can compute its location by using the estimated distances from the three nearby beacons. The distance can be estimated based on, e.g., the relative signal strength or time difference of arrivals. Analogously, in an angulation approach, a sensor node can find its location by using the angle of arrival information from three anchors.

- **Range-free localization:** Instead of estimating either the distance or angle of arrival from anchors, this approach relies on the mere presence of anchors. By figuring out what beacons are nearby, heuristics can be employed to provide a coarse-grained location estimate [3, 7].

In addition, several schemes [7, 16] have been proposed for multi-hop localization, in which the number of anchors is insufficient to directly localize all the nodes in the WSN. For example, the DV-hop algorithm [16] shown in Figure 3 uses a distance-vector flooding technique to determine the minimum hop count and average hop distance to known anchor positions. Each anchor broadcasts a packet including its location and a hop count, initialized to one. The hop-count is incremented by each node as the packet is forwarded. Each node maintains a table of the minimum

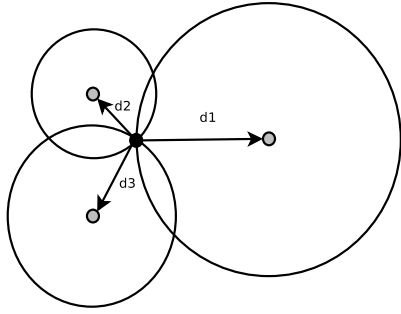


Figure 2. Triangulation-based Localization

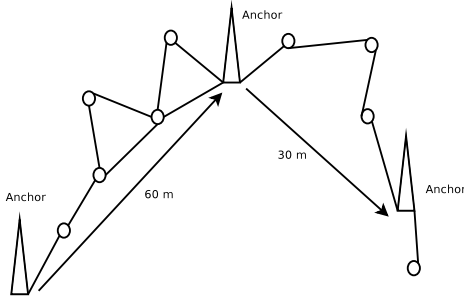


Figure 3. Multi-hop Localization

hop-count to each beacon. An anchor can use the absolute location of another anchor and the minimum hop count to that anchor to calculate the average distance per hop. The beacon broadcasts the average distance per hop, which is forwarded to every node. To compute its position via lateration, an individual node uses the average per-hop distance and its hop count to anchors whose positions are known.

2.3 Location Verification

Without verification, a malicious node can falsify its location information to compromise the basis of GR. Similar threats exist when the location information is used for other services such as access control or storage [19]. To prevent a sensor node from falsifying its location, Sastry et al have proposed a location verification scheme [21] in which a sensor node needs to send its location claim to a verifier that subsequently sends back a challenge to the node. When the node receives the challenge, it should immediately reply to the verifier, through an ultrasonic channel, with a nonce that was included in the original challenge message. To verify the location, the verifier measures the delay between the challenge and response. It compares the measured delay to the delay estimated according to the claimed location and speed of sound. However, this approach requires ultrasonic hardware and it verifies the claimed location relative to only one verifier. Moreover, an immediate response may not always be possible, e.g., due to overloads or packet losses. As

a result, honest nodes could be unnecessarily invalidated.

We have developed an alternative approach for location verification [1]. The key idea is to reverse the localization procedure of the triangulation-based methods including the ones using the Radio Signal Strength (RSS), Time of Arrival (ToA), Time Difference of Arrival (TDOA), and Angle of Arrival (AoA) discussed in Section 2.2. In our approach, a sensor node is not allowed to generate its own location estimate, but it has to transmit a localization request to the anchors in the neighborhood. This request is assumed to be received by three or more anchors. Each of the anchors produces an estimate of the distance (if the RSS or TDOA is used) or angle (if the AoA is used) based on the request received from the sensor. The anchors exchange this information with each other to securely produce a location estimate via triangulation. The location estimate is then provided to the querying sensor node with a certificate. Thus, the certified location information can be securely exchanged with other nodes, while disallowing a compromised or malicious node to spread false location information. Our approach can be used to verify a location even if it is not the primary localization algorithm used by the sensors. Thus, in the remainder of this paper, we assume that the location information is verified.

3 Threat Model

Since it is impossible to consider all security problems across varying WSN configurations, it is reasonable to tackle a secure routing problem under a certain threat model [11]. The threat model considered in this paper is as follows.

Cryptography-based Link Layer Protocol. We assume a cryptography-based link layer protocol such as [10, 18, 14] is used to support the confidentiality, integrity, and authenticity of messages exchanged between the base station, anchors, and sensor nodes.

Types of Nodes. There are three types of nodes, i.e., a base station, anchors, and sensor nodes. These nodes have different trustworthiness:

- The base station is trustworthy. This is a common assumption in WSN security [11, 18, 10, 14].
- An anchor knows its own location, for example, using a GPS. This is a common assumption for localization in WSNs as discussed in Section 2.2. Further, anchors are considered trustworthy. (Resilient routing in the presence of compromised anchors is reserved for future work.)
- Sensor nodes are not trusted unlike the base station or anchors. This is a common assumption in WSNs

[11, 18, 10, 14], because it is relatively easy for an adversary to capture and compromise sensor nodes, for example, by extracting their cryptographic keys or downloading malicious code.

Attacks and Design Goals. We aim to design a robust geographic routing protocol that can effectively handle the following attacks:

- An adversary can send an excessive number of packets to its neighbors to cause the queue overflow in the nodes. Thus, many packets from legitimate nodes can be dropped in those nodes, while other sources intending to communicate with the attacked nodes may get blocked due to the significant channel contention and overload. We aim to alleviate the potential damage of flooding attacks by favoring well-behaving nodes via rate control, scheduling, and routing.
- Blackhole and selective forwarding attacks, in which an adversary drops all or selected packets, are possible. We aim to reduce the possible damage due to blackhole/selective forwarding attacks by taking multiple paths towards the destination and track trustworthiness of forwarders based on their behavior. In this way, we can improve the probability of packet delivery even in a hostile environment such as a battle field.

In this paper, we consider no physical or MAC layer attack such as radio jamming, because these attacks cannot be addressed by location verification or secure routing. Lower level solutions such as frequency hopping are required to mitigate such attacks. Instead, we focus on designing a robust GR protocol that can alleviate the possible damage due to potential packet flooding and blackhole/selective forwarding attacks when the verified location information is given.

4 Robust Geographic Routing

In this section, we discuss our approach to secure geographic routing in which a node performs rate control, packet scheduling, and routing based on the observed behavior of nodes.

4.1 Rate Control and Packet Scheduling

An adversary can compromise several sensor nodes to send their neighbors a lot of messages authenticated using their cryptographic keys. Note that this attack cannot be handled by cryptographic approaches including [18, 10, 14], since every packet is properly authenticated. As a result, honest nodes in the vicinity can be significantly delayed due to overload and channel contention as discussed

before. The attack will be especially successful, if the neighbor simply forwards the received packets to the next hop causing cascading damage. To avoid the problem, each individual node does rate control and packet scheduling in our approach.

In WSNs, the base station usually disseminates queries specifying the sensor data of interest, the corresponding sampling frequency, and the time duration of the report. It can flood queries via an authenticated broadcast protocol such as μ TESLA [18] to allow each node to verify and identify the queries issued by the base station. Thus, a node can compute how many packets it will receive when it is involved in answering specific queries. In our *rate control* mechanism, a sensor node can defend itself against flooding attacks by dropping excessive packets received from a neighbor that violates the expected maximum rate computed above. Observe that we can only approximate the maximum rate, because a node may not receive a query from the base station, e.g., due to packet losses. Hence, we do not drop all the packets coming from a (seemingly) misbehaving node. (In some routing protocols such as directed diffusion [8], the base station reinforces the path providing the sensor data related to the query with the minimum delay. Authenticated reinforcements, if available, can help a related node to estimate the rate bound more accurately.)

In addition to dropping excessive packets, we *prioritize packets* by assigning a low priority to the remaining packets from the misbehaving node to further improve the robustness of our protocol. When 1 indicates the highest priority level, node i assigns a priority level $p_{i,j}$ to the packets coming from a one hop neighbor j :

$$p_{i,j} = \begin{cases} 1 & \text{if } r_{i,j} \leq M_{i,j} \\ 2 & \text{if } M_{i,j} < r_{i,j} \leq b_1 \\ 3 & \text{if } b_1 < r_{i,j} \leq b_2 \\ \vdots & \\ \vdots & \\ n & \text{if } r_{i,j} > b_{n-1} \end{cases} \quad (1)$$

where n is the number of priority levels, $r_{i,j}$ ¹ is the arrival rate of the packets from node j , b_k is the k^{th} rate bound used to prioritize packets based on the incoming rate, and $M_{i,j}$ is the rate bound (derived from the queries received from the base station as discussed before) at which node j is allowed to send packets to node i . If nodes i and j are simultaneously involved in multiple queries, $M_{i,j}$ is the sum of the sampling frequencies of the queries.

In our approach, there is a dedicated FIFO queue for each priority level; that is, an incoming packet with priority ℓ is appended at the end of the queue q_ℓ upon the arrival. In our approach, fixed priority is enforced between the multi-level queues. Thus, a packet in a queue can only be transmitted

¹The moving average can be taken if $r_{i,j}$ changes from time to time due to network dynamics.

after every packet in the higher priority queues, if any, has been transmitted. In addition, we control the link utilization in a prioritized manner; that is, the packets in the lowest priority queue, i.e., q_n , will be dropped first when the link utilization is higher than the specified threshold, e.g., 80%, until the utilization decreases below the threshold.

Moreover, a node selects the forwarding route in a differentiated manner. When it receives a packet from a misbehaving node, it randomly selects a single neighbor in its FS to forward the packet to. Since the packet is from the misbehaving node, our routing protocol does not necessarily forward it to the node closest to the sink. Instead, it evenly distributes the load by randomly selecting the next node from the FS. By diversifying routes, we can also pick other neighbors, which may not be selected in the original GR protocol, to observe whether or not they actually forward the packet and deliver it to the right direction. In this way, we can quickly build more trust information that may be necessary to securely forward packets to the sink or different sinks, if any, in the future by judiciously taking advantage of redundancy in a dense network. In contrast, a node forwards a packet received from a well-behaving node to multiple next hops that are trustworthy and closest to the sink to support the reliable delivery of the packet. Overall, we favor the packets arrived from well-behaving nodes via rate control, scheduling, link utilization control, and routing. A more detailed discussion of multi-path routing is given next.

4.2 Trust-based Multi-Path Geographic Routing

In this section, we discuss how to verify whether or not a node is forwarding a received packet, build the trust information based on the forwarding check, and actually forward the packet via trust-based multi-path routing.

4.2.1 Forwarding Verification

To circumvent blackhole/selective forwarding attacks, we need to ensure that intermediate nodes actually forward packets of which they are in charge (or provide a feedback indicating the reason for packet dropping). One of possible approaches is overhearing. When a sensor node A sends a packet to node B, which is one of its one hop neighbors, A waits for the acknowledgment (ACK) from B. At the same time, A overhears B to observe whether or not B forwards the packet. Although overhearing can be easily done due to the omni-directional nature of radio communications, this basic verification scheme may not be perfect for two primary reasons:

- Node A may miss node B's transmission due to a collision with another packet; and

- Node B may forward the packet, but to a node in the wrong direction or even to a non-existing node. Since A does not know B's neighbors, it is not able to determine that B is misdirecting the packet.

To address the first case, a node needs to monitor a neighbor's behavior for multiple packets to evaluate its trustworthiness. To improve the correctness of forwarding verification based on overhearing, we consider the following approaches:

- Node A can query an anchor about the location of the destination to which B is forwarding a packet to determine whether it exists and it is closer to the destination. If B forwarded a packet to node C and C exists, A can cache C's ID and location. In this way, A can incrementally build the two hop neighbor information. Hence, it can perform the forwarding verification without querying the anchor when B forwards another packet to C.
- Optionally, trust information can be built more quickly by allowing mutually trusting nodes to periodically exchange the reputation about their neighbors in a cryptographically secure manner to form trusted cliques. In this way, an individual node can get more trustworthy information about its neighbors derived from the broader perspectives of its trusted neighbors. Since this is an optional feature, sensor nodes can be configured to only rely on its own trust information if the environment, e.g., a battle field, is highly hostile.

Since the forwarding verification may not be 100% accurate, we take a probabilistic approach to selecting the next hop nodes to forward a packet towards the sink as follows.

4.2.2 Trust Management

The basic idea of our trust management scheme is to favor honest nodes by giving them the credit for each successful packet forwarding, while penalizing suspicious nodes that supposedly lie about or exaggerate their contribution to routing. If a node lies about its location, it is immediately excluded from the FS. In the worst case, the FS of a node under a severe attack may become empty. In this case, the node has to rely on the face routing for void traversal, similar to [12]. We take this approach, since the trustworthiness of the location information is essential for geographic routing. Further, this approach discourages potential attackers to attempt to provide false location information to disrupt geographic routing. Thus, (selective) packet dropping due to more stealthy routing disruption or poor wireless communication quality is the main reason for penalty. Overall,

an honest node with good link quality towards the destination will stay longer in the FS to support secure geographic routing.

A node monitors the behavior of the one hop neighbors to which it forwards packets. Specifically, we define the trust level of a neighbor node to be between 0 and 1 to indicate no trust and full trust, respectively. When node i 's location has been verified, its trust level T_i is set to a certain initial value, e.g., 0.5.

If the source detects that a neighbor node i (\in FS) has successfully forwarded a packet towards d , it will increase the trust level of node i :

$$T_{i_{new}} = \begin{cases} T_i + \delta t & \text{if } T_i + \delta t \leq 1; \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

where δt is the specified step size, e.g., 0.01.

As discussed before, an adversary in the FS can drop the packet or forward it to a node in a wrong direction, while returning an ACK. By overhearing, s can check whether a neighbor i has actually forwarded the packet towards d , and thereby, confirming the trustworthiness of the ACK that it receives. Specifically, when a node i is suspected to disrupt routing by returning an ACK without properly forwarding the packet, its trust level is decreased:

$$T_{i_{new}} = \begin{cases} T_i - \Delta t & \text{if } T_i - \Delta t > 0; \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where Δt is the predefined penalty for each suspicious behavior. By exchanging trust information with a trustworthy node j , s could further infer i 's trustworthiness.

Note that we do not immediately remove a node from the FS when it is suspicious of dropping a few packets, because it may be honest but currently suffer, for example, a transient congestion. When the node recovers from the transient network problem, it can contribute to secure geographic routing, while improving its trust level. If a node suffers chronic network problems or little remaining energy, it will eventually be removed from the FS.

4.2.3 Multi-Path Routing

We intend to forward packets from well-behaving nodes, which respect the rate bound, to the honest next hops that have relatively high trust values. The pseudo code of our protocol is as follows.

1. When a source s wants to transmit a packet towards a destination d for the first time, it establishes a shared secret with a local anchor via a cryptography-based link layer protocol such as [18, 10, 14]. It also queries the anchor to get the verified geographic information of the neighbors in a certain range, e.g., twice its radio range, if it does not have the information already.

(Alternatively, it can incrementally build the two hop neighbor information as discussed before.) The location information can be encrypted and authenticated using the shared key.

2. The source broadcasts a transmission initiation packet, which can be an authenticated RTS (Request To Send) packet including the source and destination locations.
3. Upon receiving the initiation packet, a neighbor verifies the authenticity and integrity of the received packet and adds the source and destination information to the routing table. In addition, it returns an authenticated CTS (Clear To Send) packet to s .
4. The source s verifies the authenticity of the CTS packet received from the neighbor. If the verification is successful, it adds the ID and location information of the neighbor to the routing table unless it already exists.
5. Compute the probability P_i of forwarding a packet to a one hop neighbor $i \in$ FS that is the set of nodes that are geographically closer to d than s is and its trust level T_i is greater than or equal to the threshold θ_1 . Specifically, we set $P_i = T_i / \sum_{i=1}^N T_i$ where N is the cardinality of the FS. (A detailed description of T_i initialization and management is given in Section 4.2.2.) Given $\{P_1, P_2, \dots, P_N\}$, s independently selects k neighbors in FS to which it will forward the packet where k is the required level of redundancy. Specifically, we use the roulette wheel selection technique [6] for node selections, since it has no bias in selection, while directly considering the candidate fitness, i.e., the trust level of a node in our approach.
6. The source selectively floods the packet to the k neighbors and overhears them, while waiting for the corresponding ACKs. If s overhears a neighbor forward a packet, it checks whether the packet has been forwarded to a legitimate location by looking up its cache or querying the anchor, if the needed information is not cached. According to the verification results, it also adjusts the trust level of the neighbor. (Forwarding verification can be performed less frequently as the energy level decreases. A more detailed discussion is given in Section 4.3.)
7. If s finds a node i whose trust level $T_i \geq \theta_2$ where $\theta_1 < \theta_2$, it periodically exchanges the trust information with node i in a cryptographically secure manner to build more global trust information that can further improve the source's own trust information and vice versa. (This is an optional step as discussed before.)
8. When node i receives the packet, it becomes a new source and recursively applies this procedure to forward the packet towards d .

In summary, our approach prioritizes incoming packets and transmits outgoing packets in a differentiated manner based on the behavior of the packet sources. Packets from misbehaving nodes trying to flood the neighbors are more likely to be dropped or lost during the transit in proportion to the degree of flooding. At the same time, our protocol supports the reliable delivery of the packets from well-behaving nodes via trust-based multi-path routing, in which next hops are selected according to the trust that is measured based on their packet forwarding history.

4.3 Security Analysis and Tradeoffs

The trust management algorithm is fully distributed in that a node can manage the trust levels on its own. In a relatively benign environment, e.g., a smart building, the trust information can be exchanged between trusted nodes with care. Thus, we can balance between the more global trust information and potential security risk due to information exchanges.

The value of the threshold used to compute the FS determines the responsiveness of our protocol to a possible routing disruption attack. If the threshold of candidate selection, i.e., θ_1 in Section 4.2, is high, a suspicious node can be excluded earlier; however, a node with no malice could be excluded prematurely due to transient network problems such as a wireless network congestion. Thus, it is necessary to derive a good threshold value that can balance the speed of suspicious node exclusion and potential false positives. In general, we believe there is no single threshold value that can optimize the tradeoff for every application, but one has to select an appropriate value, for example, by using a higher threshold in a more hostile environment.

We consider two approaches to reducing the energy consumption for forwarding verification. In the first approach, a node can *periodically* perform the verification rather than doing it for every packet, while increasing the period as the neighbors are being monitored for more packets. Alternatively, a node can *randomly* perform the verification, while reducing the verification probability as the energy level decreases for forwarding many packets to the neighbors and monitoring their behavior. The latter can be more resilient to stealthy attacks in which malicious neighbors properly forward packets only at the verification periods.

In addition, there can be several design choices with respect to Δt and δt . When $\Delta t > \delta t$, for example, we can decrease the time period during which a compromised node in the FS to subvert the protocol. This is a conservative approach more applicable to trust management in a hostile environment. Alternatively, it is also possible to manage the trust in a more optimistic manner by setting, for example, $\Delta t \leq \delta t$ when the environment is considered relatively benign. Further, the absolute size of Δt or δt determines the

trade-off between the speed of trustworthiness convergence and false positives/negatives.

5 Experimental Evaluation

We have implemented a subset of the defensive features supported by our robust geographic routing protocol discussed in Section 4. Specifically, we simulate the multi-path routing and trust management based on overhearing in the network simulator *ns2* version 2.29 [25]. The preliminary results show that our approach can significantly improve the packet delivery ratio in the presence of blackhole attacks compared to the well known GPSR protocol [12]. In the future, we will also implement our rate control, packet scheduling, and link utilization management schemes to further improve the delivery ratio in the presence of flooding attacks. The details of the experimental settings and results are discussed next.

5.1 Design of Experiments

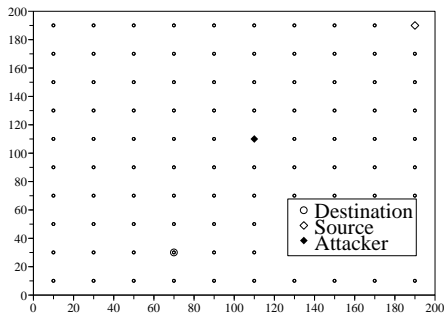
The trust management scheme is implemented by extending the GPSR algorithm [12]. We also extended the IEEE 802.11 protocol to support overhearing needed in our approach. The extended version is called RGR (Resilient Geographic Routing), while the original GPSR is called InS GPSR (Insecure GPSR) in the remainder of this section.

Radio Range	30 m
Bandwidth	2Mbps
Data Payload	64B
Packet Size	158B
Data Rate	2 packets/s
Queue Length	100 packets
Hello Period	5 s
Traffic duration	200 s
T_i initial value	0.5
Δt	0.1
Transmit Power	0.5 Watt
Receiving Power	0.2 Watt

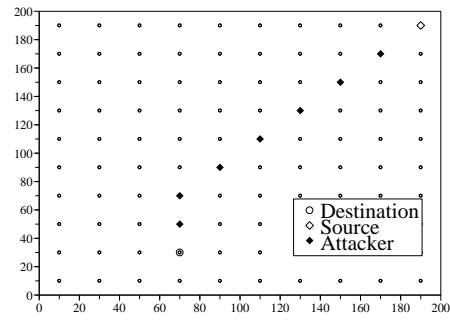
Table 1. Simulation parameters

In our experiments, 100 sensors are regularly deployed in 10x10 grids covering an area of 200x200 square meters, in which each node is located in the center of each grid. A fixed data sink (or destination) is located at the bottom. Table 1 summarizes the key simulation parameters.

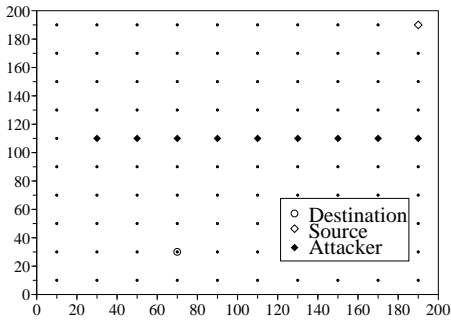
To consider different attack models, we use the five different scenarios shown in Figure 4. In the scenario 1, only one attacker resides on the shortest path from the source to destination constructed by GPSR. In the scenario 2, all the



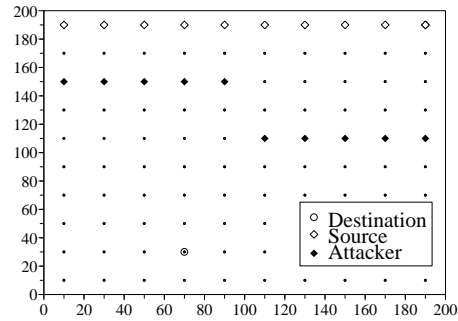
(a) Scenario 1



(b) Scenario 2

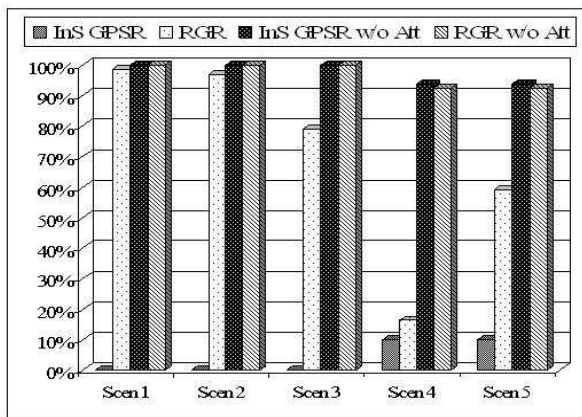


(c) Scenario 3

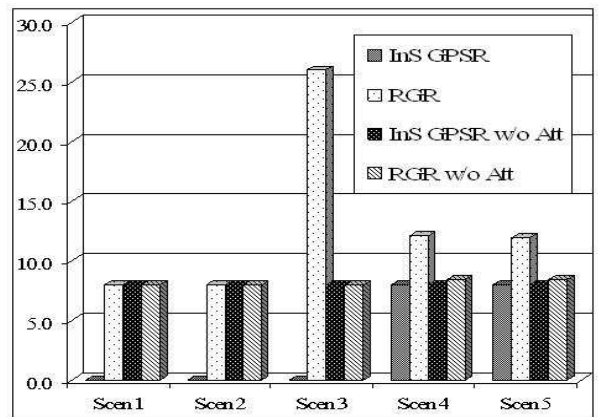


(d) Scenarios 4 and 5

Figure 4. Networking Settings for the Simulation Study



(a) Delivery Ratio



(b) Path Length of the Received Packets

Figure 5. Basic Study

nodes constructing the shortest path are attackers. In the scenario 3, the nine attackers form a wall across the network and try to separate the source and destination. For the scenarios 1 – 3, we fix δt as 0.01.

The scenarios 4 and 5 share the same network structure. In Section 5.2, the scenarios 4 and 5 use the different δt values, i.e., 0.01 and 0.02, respectively. In Section 5.3, the scenario 5 is further varied in terms of δt . We also vary the data rate and number of the sources to thoroughly evaluate the performance impact of the trust management under the different communication settings.

5.2 Basic Study

In this section, we compare the performance of RGR to that of InS GPSR.

5.2.1 Delivery Ratio

Figure 5(a) shows the delivery ratio achieved in the different scenarios. In the scenarios 1 to 3, InS GPSR completely fails if any attackers are on the path from the source to destination. As a result, no packet is received by the destination. RGR’s high delivery ratio in the scenarios 1 – 3 suggests that RGR can find the forwarding path from the source to destination, if there exists one with non-malicious nodes after an initial period in which the trust levels are estimated.

In Figure 5(a), the delivery ratio of RGR in the scenario 4 is nearly as low as that of InS GPSR, which is surprising. A further examination has shown that RGR punishes *unintentional* packet droppings when $\delta t = 0.01$ that is lower than necessary to sustain an effective path under high contention. To verify this argument, in the scenario 5, we have performed the same experiment with $\delta t = 0.02$. In this experiment, we observe that RGR achieves a considerably higher delivery ratio due to the lower penalty faced by network anomalies such as congestion. Generally, in the absence of attackers, RGR performs almost identically to GPSR without compromising the scalability and efficiency of geographic routing, while significantly improving the delivery ratio under attack.

5.2.2 Path Length

The path length is shown in Figure 5(b). In the scenarios 1, 2 and 3, when there is no attacker, any packet in RGR takes the same length of the path taken by InS GPSR. This means RGR is able to find the shortest path measured in the number of hops. In the scenario 3, any successfully received packet has to go around the “wall” of attackers increasing the path length. RGR recognizes the wall as a void due to the overhearing and trust-based route selection. Thus, it applies the perimeter routing to forward packets around the wall. In contrast, in InS GPSR, nodes do not monitor

the neighbors’ behavior; therefore, they cannot identify the packet dropping wall.

In Figure 5(b), the path lengths observed in the scenarios 4 and 5 are similar. The average path length of GPSR does not increase even under attack, because the routing decision of GPSR is only based on the connectivity and insensitive to security. If there is no attacker, the path length of RGR is slightly longer than GPSR. The path length may have been increased due to the penalty given to certain hot-spot nodes. Any packet dropping, e.g., because of a queue overflow, may cause packets to be routed away from the optimal forwarding node. Regardless of the delivery ratio for different δt values, RGR’s path length does not change significantly from the scenario 4 to scenario 5.

5.3 Effect of Trust Adjustment Parameters

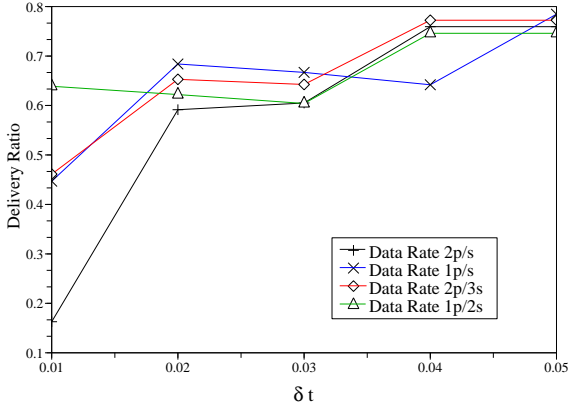
In section 5.2, adjusting the value of δt has resulted in the considerably different performance of RGR in the scenarios 4 and 5. Thus, in this section, we analyze the performance of RGR for several δt values and network loads.

5.3.1 Impact of Trust Increment Parameter δt

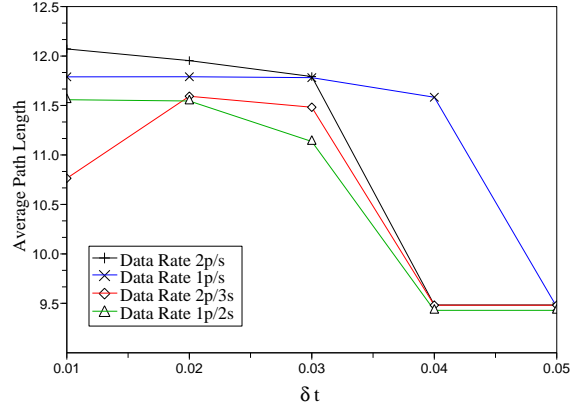
In this subsection, we measure RGR’s performance for different δt values increasing from 0.01 to 0.05 by the step size 0.01, while fixing $\Delta t = 0.1$. Figure 6(a) shows the impact on the delivery ratio. Despite the different data rates, the delivery ratio increases linearly as δt increases, since the hop-spot intermediate nodes receive less penalty for congestion. Further, as shown in Figure 6(b), the path length decreases as δt increases. However, simply increasing δt may not always improve the delivery ratio, since some attackers can selectively forward data packets, while keeping their trust level high. From this experiment, we learn that we need to avoid using too small a δt value (more accurately $\frac{\delta t}{\Delta t}$) which would give unnecessary penalties due to congestion or collision related packet losses.

5.3.2 Impact of Traffic

In this subsection, we study the impact of the traffic load on the performance of RGR. To this end, we consider the scenario 5 and set $\Delta T = 0.1$ and $\delta t = 0.02$, while increasing the number of sources from 1 to 10. We also study four different data rates. By increasing the network load, more packets can be lost due to congestion possibly polluting the trust estimates. Figure 7 shows the delivery ratio. When the number of data sources is small, the delivery ratio is improved as the data rate increases, because a relatively small portion of packets are needed for initial trust training. With more data sources, a lower data rate generally shows the better performance due to less congestion and fewer errors in the trust estimation.



(a) Delivery Ratio



(b) Path Length

Figure 6. Impact of δt

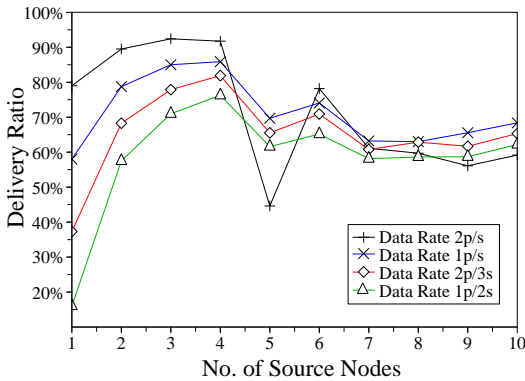


Figure 7. Impact of Traffic on the Delivery Ratio

6 Related Work

Wireless sensor networks are exposed to numerous security attacks. Karlof et al. [11] discuss the possible routing disruption attacks and countermeasures. They pointed out that false location claims can seriously disrupt a geographic routing protocol, while suggesting multi-path routing as a countermeasure against selective forwarding attacks. However, a detailed approach was not discussed. There are a number of challenging security problems related to location verification, localization, and routing that cannot be directly addressed by cryptography-based link layer security protocols such as [18, 10].

Localization has been well studied [17, 22, 24, 2, 7, 15,

3, 16]; however, most existing approaches do not address security. Sastry et al. [21] propose a secure location verification protocol. When a node claims its location, a single verifier can check whether or not the location claim can be trusted by leveraging the time difference between the radio and ultrasonic signal arrivals, which is hard for an adversary to subvert. We have developed a different approach [1] for location verification that does not require an ultrasonic channel, while providing more accurate full location verification, rather than relying on the distance to a single verifier [21].

Lazos et al. [13] address a complementary problem—a secure range-independent localization problem. Their protocol can enable a sensor node to securely derive its location using trusted anchors. This protocol considers attacks on the localization mechanism to cause nodes to have erroneous location information, but it does not prevent a misbehaving node from providing false estimates of its own location to its neighbors.

Geographic routing protocols such as Greedy Perimeter Stateless Routing (GPSR) [12] and Geographic and Energy Aware Routing (GEAR) [27] can leverage the geographic locations of the source and destination for efficient routing. In GPSR, a node greedily forwards a packet to the neighbor geographically closest to the destination among its one hop neighbors. When there is a void in the network, GPSR routes packets around the hole. GEAR is an energy aware geographic routing protocol. To avoid quickly draining the energy of the node closest to the destination, it considers the remaining energy in addition to the geographic location when it selects the next node. Geographic Probabilistic Routing [20] assigns the packet forwarding probability to each neighbor based on its geographic location, residual

energy, and link reliability to further optimize the performance and energy efficiency. However, these geographic routing protocols can be compromised by an adversary lying about its location. The adversary can attract a lot of traffic by claiming several geographic locations, a high energy level, and link quality, while selectively dropping the packets. We propose to prevent the Sybil attack by location verification, while monitoring the behavior of the neighbors to detect if a compromised or malicious node, if any, tries to subvert the geographic routing.

ARRIVE [9] is a robust routing protocol applicable to wireless sensor networks with a tree-like topology. It overhears the behavior of the neighboring nodes to make probabilistic packet forwarding decisions. To overcome the unreliability of wireless communications, a node forwards a packet to not only a parent but also its neighbors with the reputation higher than the threshold. Different from ARRIVE, we consider the geographic routing problem, while taking advantage of verified location information for routing. Also, we can alleviate the impact of flooding attacks and perform forwarding verification for trust-based routing, while allowing trustworthy nodes to exchange the trust information between them to derive a more global view.

Virendra et al. [26] propose a novel approach to quantifying trust in mobile ad-hoc networks. Our work is analogous to theirs in that our approach monitors the behavior of packet arrival and relay patterns in the context of geographic routing. MIAMI [23] addresses the process of management (PoM) problem. Given a data packet, the classifier in a node takes the data object to apply a consistency check to determine whether or not data can be considered reliable. Their work is complementary to ours in that our work can be further extended by considering the consistency of sensor data in addition to monitoring the routing behavior of neighbors. As a result, resilient routing can be supported, while the consistency of the forwarded data are checked.

7 Conclusions and Future Work

While security in WSNs and ad hoc networks has been well studied, most existing work have focused on traditional routing protocols. The nature of GR makes it vulnerable to a different set of attacks and require specialized solutions for securing them. To address the problem, we explore the problem of resilient geographic routing. Even if location information is verified, nodes may still misbehave, for example, by sending an excessive number of packets or dropping packets. To dynamically avoid untrusted paths and continue to route packets even in the presence of attacks, the proposed solution uses rate control, packet scheduling, and probabilistic multi-path routing combined with the trust-based route selection. We discussed the proposed approach in detail, outlining alternative choices. We

considered possible attacks and defenses against them. In addition, we compared the performance of our resilient geographic routing protocol to a well-known geographic routing protocol. There are several open research issues that remain to be addressed. First, we have not considered the implications of range-free/multi-hop localization algorithms such as [3, 7, 16] in terms of secure localization or location verification. Securing range-free localization is important, because it is necessary when the density of anchors is not high enough to support range-based localization. The virtual coordinate routing protocol [4] is interesting in that it can conceptually emulate geographic routing without requiring physical location information. We will investigate possible attacks to virtual coordinate routing and develop countermeasures. Also, we will further extend our secure geographic routing protocol by improving the rate control, scheduling, route selection, and other fundamental techniques needed for routing in wireless sensor networks.

References

- [1] N. Abu-Ghazaleh, K. D. Kang, and K. Liu. Towards Resilient Geographic Routing in Wireless Sensor Networks. In *1st ACM Workshop on QoS and Security for Wireless and Mobile Networks (Held in Conjunction with ACM/IEEE MSWiM 2005)*, Oct. 2005.
- [2] P. Bahl and V. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *INFOCOM*, 2000.
- [3] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less Low-Cost Outdoor Localization for Very Small Devices. *IEEE Personal Communication*, 2000.
- [4] A. Caruso, S. Chessa, S. De, and A. Urpi. GPS Free Coordinate Assignment and Routing in Wireless Sensor Networks. In *IEEE Infocom*, 2005.
- [5] R. Fonseca, S. Ratnasamy, J. Z. and Cheng Tien Ee, D. Culler, S. Shenker, and I. Stoica. Beacon vector routing: Scalable point-to-point routing in wireless sensor networks. In *NSDI'05*, May 2005.
- [6] D. Goldberg. *Genetic Algorithm in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1989.
- [7] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher. Range-Free Localization Schemes for Large Scale Sensor Networks. In *MobiCom'03*, 2003.
- [8] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networks (MobiCOM 2000)*, 2000.
- [9] C. Karlof, Y. Li, and J. Polastre. ARRIVE: Algorithm for Robust Routing in Volatile Environments. Technical Report UCB//CSD-03-1233, University of California at Berkeley, 2003.
- [10] C. Karlof, N. Sastry, and D. Wagner. TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. In *ACM SenSys*, 2004.

- [11] C. Karlof and D. Wagner. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. In *Sensor Network Protocols and Applications*, 2003.
- [12] B. Karp and H. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *MobiCom*, 2000.
- [13] L. Lazos and R. Poovendran. SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks. In *the ACM Workshop on Wireless Security*, 2003.
- [14] D. J. Malan, M. Welsh, and M. Smith. A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography. In *IEEE SECON*, 2004.
- [15] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network. In *2nd International Workshop on Information Processing in Sensor Networks (IPSN 03)*, 2003.
- [16] D. Niculescu and B. Nath. Ad Hoc Positioning System (APS). In *Global Telecommunications Conference*, volume 5, 2001.
- [17] D. Niculescu and B. Nath. Ad Hoc Positioning System (APS) Using AOA. In *INFOCOM*, 2003.
- [18] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar. SPINS: Security Protocols for Sensor Networks. In *MobiCom*, 2001.
- [19] S. Ratnasamy, D. Estrin, R. Govindan, B. Karp, S. Shenker, L. Yin, and F. Yu. Data-centric storage in sensor networks. In *Proceedings of the First ACM SIGCOMM Workshop on Hot Topics in Networks*, Oct. 2002.
- [20] T. Roosta, M. Menzo, and S. Sastry. Probabilistic Geographic Routing in Ad Hoc and Sensor Networks. In *International Workshop on Wireless Ad-hoc Networks*, 2005.
- [21] N. Sastry, U. Shankar, and D. Wagner. Secure Verification of Location Claims. In *the ACM workshop on Wireless security*, 2003.
- [22] A. Savvides, C. Han, and M. Srivastava. Dynamic Fine Grained Localization in Ad-Hoc Sensor Networks. In *Mobicom*, 2001.
- [23] W. Trappe, Y. Zhang, and B. Nath. MIAMI: Methods and Infrastructure for the Assurance of Measurement Information. In *2nd International VLDB Workshop on Data Management for Sensor Networks*, 2005.
- [24] S. Čapkun, M. Hamdi, and J.-P. Hubaux. GPS-free positioning in mobile ad-hoc networks. In *HICSS*, 2001.
- [25] USC ISI. Network simulator 2, 2005.
- [26] M. Virendra, M. Jadliwala, M. Chandrasekaran, and S. Upadhyaya. Quantifying Trust in Mobile Ad-Hoc Networks. In *International Conference Integration of Knowledge Intensive Multi-Agent Systems*, 2005.
- [27] Y. Yu, R. Govindan, and D. Estrin. Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks. Technical report, Computer Science Department, UCLA, 2001.