Towards Resilient Geographic Routing in WSNs

Nael Abu-Ghazaleh, Kyoung-Don Kang and Ke Liu
Department of Computer Science
State University of New York at Binghamton
{nael,kang,kliu}@cs.binghamton.edu

ABSTRACT

In this paper, we consider the security of geographical forwarding (GF) – a class of algorithms widely used in ad hoc and sensor networks. In GF, neighbors exchange their location information, and a node forwards packets to the destination by picking a neighbor that moves the packet closer to the destination. There are a number of attacks that are possible on geographic forwarding. One of the attacks is predicated on misbehaving nodes falsifying their location information. The first contribution of the paper is to propose a location verification algorithm that addresses this problem. The second contribution of the paper is to propose approaches for route authentication and trust-based route selection to defeat attacks on the network. We discuss the proposed approaches in detail, outlining possible attacks and defenses against them.

Keywords

Location Verification, Secure Routing, Sensor Networks

1. INTRODUCTION

Wireless sensor networks (WSN) hold the promise of revolutionizing sensing across a large number of disciplines. Geographic Forwarding (GF) is an attractive approach for routing in WSNs because of its low overhead and localized interactions. In GF, nodes exchange their location information with neighbors; forwarding decisions are made by selecting a neighbor that is closer to the destination. Thus, GF relies on an underlying localization mechanism that allows the sensor nodes to determine their geographic location [3, 6, 16]. GF and localization are discussed in more detail in Section 2.

While a number of studies consider security in multi-hop wireless networks such as WSNs and Ad hoc networks, they have mostly focused on conventional routing protocols. There are fundamental differences between GF and traditional routing algorithms: GF is distance-based, rather than connectivity-based. Nodes interact only with their neighbors, taking localized forwarding decisions. Furthermore,

the use of location, rather than node IDs introduces additional security concerns. For these reasons, there are a number of unique attacks that are possible on geographic forwarding. In this paper, we outline some of the important attacks on GF, and present solutions to them. We discuss the threat model in more detail in Section 3. Very little work has studied the issue of securing GF and the underlying localization mechanisms; an overview of related work is presented in Section 6.

One of the attacks on GF is possible because no oversight mechanism exists for verifying that a node is at the position it is claiming; misbehaving nodes can falsify their location information. This can lead to the use of suboptimal routes, misrouting of packets, or a wormhole attack [18]. The first contribution of the paper is to propose a location verification algorithm that addresses this problem. The proposed approach has several advantages over another recently proposed location verification algorithm [22]. We present the location verification algorithm and discuss possible attacks on it and defenses to them in Section 4.

While location verification allows nodes to have confidence in the location of their neighbors, thereby preventing many straightforward attacks on routing, other attacks on routing remain possible. For example, a node may simply drop packets it is tasked with forwarding. The second contribution of the paper is to propose approaches for increasing the resiliency of GF to attacks on routing. Specifically, we propose to use probabilistic multi-path, with trust-based route selection, to dynamically avoid untrusted paths and continue to route packets in the presence of attacks. We discuss the proposed approach in detail, outlining possible attacks and defenses against them in Section 5. Finally, we present concluding remarks and future research in Section 7.

2. BACKGROUND

This section presents background information necessary to understand the security issues in GF. It first overviews Geographic Forwarding (GF) and then localization.

2.1 Geographic Forwarding

Geographic forwarding is a greedy routing algorithm based on geography. More specifically, a forwarding node relays an incoming data packet to the neighbor who is the nearest to the destination among its one-hop neighbors. More generally, for a given node, all its neighbors nearer to the destination are considered the candidate *forwarding set* (FS) for that destination.

Geographic forwarding is attractive since it requires that
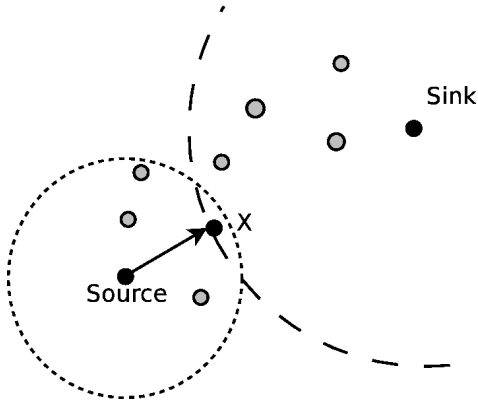
Figure 1: GF example: X is source's nearest neighbor to sink

Figure 2: Voids: X is a local minima

Figure 3: Localization Algorithm: triangulation
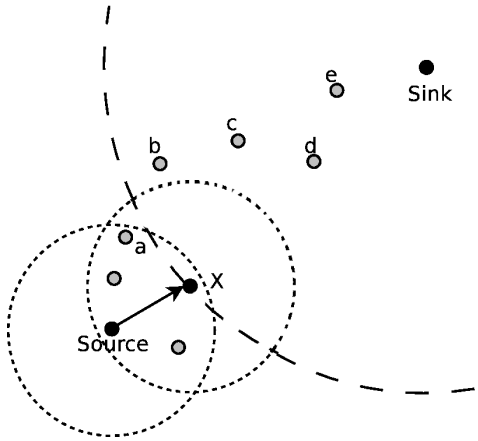
nodes only maintain the location of their one-hop neighbors. Routing decision can be made locally and dynamically. An example is shown in Figure 1.

GF does not always succeed. When the forwarding node is the nearest to the destination among all its one-hop neighbors, it cannot forward the incoming data packet further; the packet is stuck in a local minimum where FS is empty (e.g., Figure 2). In such a case, typically a complementary mechanism based on face routing [10] is used to route around the void. We do not consider the face routing mechanism in this paper.

2.2 Triangulation-based Localization

Sensor network nodes are often not equipped with their own GPS devices. Moreover, in indoors settings, GPS is not available. The alternative employed in many sensor networks is to conduct a localization algorithm where a node calculates its location relative to other nodes in the network. Localization algorithm alternatives include:

- Triangulation: Location determined using trigonometry (lateration or angulation). In the network, some anchor nodes equipped some positioning device such as GPS have their accurate location information. These anchor nodes periodically broadcast their location information to all other nodes in the network. Later-
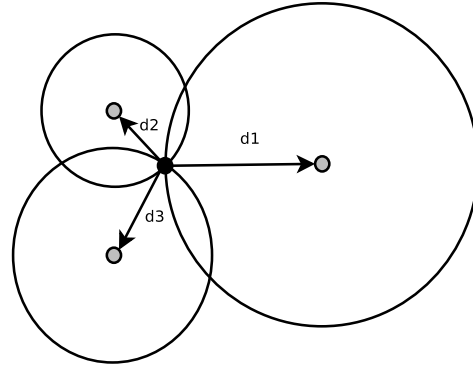
ation [16] is the calculation of position information based on distance estimated from the anchors. A 2D position requires three distance measurements. An example is shown in Figure 3, where a node uses the estimate of its distance from 3 beacons to compute its location. Several approaches are used to estimate distance such as relative signal strength, and time difference of arrival. Angulation uses angle of arrival information and also requires angles from three known anchors in order to localize.

- Proximity-based or Range-free Localization: Instead of relying on sophisticated approaches to measure either 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 estimate of location [3, 6].

- Other Approaches: Approaches such as scene analysis and dead reckoning have also been employed. In scene analysis, observed features are used to infer location using a precomputed map. In the differential scene analysis algorithms, the differences between successive scenes are compared to each other to calculate location. This kind of localization algorithm requires a compiling database which is not available for most sensor network applications. In dead reckoning, the initial location of a mobile is known, and motion sensors (accelerometer) are used to track the velocity. The velocity is used to estimate distance using dead-reckoning to interpolate between velocity measurements.

In this paper, we mostly focus on the first two approaches. Several schemes for multi-hop localization, where there are insufficient beacons to localize all nodes in the network directly, have been proposed [6, 16]. For example, DV-hop uses a distance-vector flooding technique to determine the minimum hop count and average hop distance to known anchor positions [16]. Each anchor broadcasts a packet with 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 minimum hop-count distances to each beacon. A beacon can use the absolute location of another beacon along with the minimum hop count to that beacon to calculate the average distance per hop. The beacon broadcasts the average distance per hop, which is forwarded to all nodes. Individual nodes use the average

distance per hop, along with the hop count to known beacons, to calculate their local position using lateration. An example of DV-hop is shown in Figure 4. The security of multi-hop localization algorithms is a topic for future work.
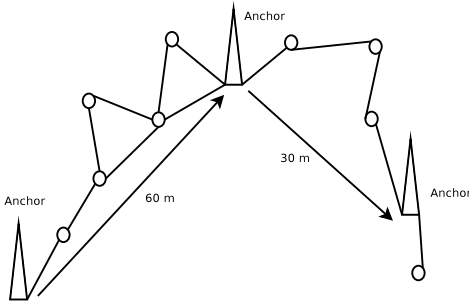

Figure 4: Multi-hop Localization

3. THREAT MODEL AND ASSUMPTIONS

It is impossible to consider all security problems across the varying wireless sensor network system configurations. Therefore, we make the following assumptions.

- There are two types of nodes: anchor nodes and sensor nodes. An anchor node is assumed to know its own location, for example, using the GPS (Global Positioning System). We assume that anchor density is sufficient to allow most sensor nodes to localize with anchors in one hop range. For the most part, we also assume anchors are trusted and they can communicate with each other in a secure manner; however, we do consider the effect of attacks on anchors.

- Each anchor or sensor is associated with a unique ID which is not compromised.

- Although either a secret key or public key system can be used for our approach, we assume an efficient public key system, such as *ecTinyos* [12] available in MICA2 motes, is used. Before the deployment, each anchor or sensor node is assigned a unique private and public key pair. Further, public keys are distributed among the anchors and sensors. Once a trusted geographic route from the source to destination is found, the two communication ends can exchange a session key for more efficient communications similar to other approaches [18, 8]. Therefore, we assume it is possible to support the message confidentiality, integrity, and authenticity using the public key or shared key shared between two communication parties.

- Sensor nodes are not trusted. We assume an adversary can capture and compromise sensor nodes, for example, by extracting their cryptographic keys or downloading malicious code.

- No physical or MAC layer attack, e.g., radio jamming, is considered. These attacks cannot be addressed by secure localization or routing; lower level solutions such as frequency hopping are required to assist in mitigating such attacks.

- Sybil attacks, in which a compromised or malicious node can claim several locations, are possible. A key

objective of our secure localization scheme is to provide a cost-effective countermeasure against Sybil attacks. By verifying the location information, we can avoid attacks trying to disrupt geographic routing by claiming false locations.

- Blackhole and selective forwarding attacks, in which an adversary drops all or selected packets, are possible. In addition to avoiding Sybil attacks by using verified location information, we aim to reduce the possible damage due to packet dropping attacks by taking multiple and relatively trustworthy paths toward the destination. In this way, we can improve the probability of packet delivery even in a hostile environment, e.g., a battle field.

- We assume the network is dense enough such that a sensor node has several one hop neighbors in its radio range. Hence, multipath routing for resiliency is possible.

4. LOCATION VERIFICATION

A unique threat that occurs in geographic forwarding (GF) is that of neighboring nodes falsifying their location in an attempt to misroute packets or route them through inefficient paths. Typically, neighboring node location information is obtained by direct exchange with these neighbors. With no verification mechanism in place, a node can falsify its location information thereby compromising the basis of GF and allowing attacks such as Sybil attack. Similar threats occur if the location information is used for other services such as access control or storage [20].

4.1 Location Verification Problem Definition

Localization is typically performed by techniques such as triangulation. In such a technique a sensor node receives beacon transmissions from anchors that allows it to estimate its distance, or angle, from them. With three such measurements, a node can calculate its location using a geometric computation. Sensors are trusted to perform the above computation and calculate their location. There is no oversight mechanism to ensure that the sensor is performing the calculation correctly, and reporting a correct location to its neighbors. Absent such a mechanism, it is possible for a misbehaving node to disrupt geographic forwarding by advertising false locations.

Recognizing the above problem, Sastry et al proposed a localization verification scheme where the location estimate produced by a node is challenged by a verifier [22]. The receiver replies immediately with a nonce included on the challenge using an ultrasonic channel. The computed delay is compared against the estimated delay from the claimed location to verify the location. However, this approach requires specific physical hardware and allows verification only relative to a single verifier. Furthermore, the requirement of immediate response may not be possible in all instances causing unnecessary invalidations under normal operation.

In this paper, we propose an alternative approach for location verification. We assume a network with mostly trusted anchors which serve as localization and location verification nodes. Such conditions are present, for example, in networks that carry out direct localization from nearby anchors (as opposed to multi-hop localization [14]). Later, we relax this assumption and outline the challenges that arise when

anchors are not fully trusted.

The key idea in our scheme is to reverse the sense of the localization such that the non-trusted sensor nodes are not responsible for generating their own location estimate. The scheme can be applied to triangulation based localization approaches including those based on Radio Signal Strength (RSS), Time of Arrival (ToA), Time Difference of Arrival (TDOA) and Angle of Arrival (AoA). In the proposed scheme, localization occurs by having sensors "transmit" a localization request. This request is received by multiple (3 or more) anchors. The anchors each produce an estimate of distance (if RSS or TDOA is used), or angle (if AoA is used) based on the received transmission from the sensor. They exchange this information with the other anchors to produce a location estimate securely via triangulation.

The location estimate can then be provided to querying nodes, or passed back to the sensors along with a certificate for exchange of certified location with other nodes in the network. The non-trusted sensor nodes are not a part of generating their location estimate, which is instead derived from the detected transmissions based on its actual location. We note that the proposed approach can be used to verify locations even if it is not the primary localization algorithm used by the sensors.

4.2 Possible Attacks

In the remainder of this section, we overview possible attacks on the proposed secure localization approach and defenses to them.

4.2.1 Localization Broadcast Manipulation

While a node is not involved in its own localization in the proposed scheme, it may attempt to influence it by exploiting the underlying localization mechanism. For example, when RSS is used distance is estimated based on measured signal strength. A node may attempt to cheat by transmitting at a higher power to appear closer than it is, or lower power to appear farther than it is. Here, we assume that anchors are trusted, and consider the possibility of their compromise as a separate attack.

If the misbehaving node, M, sends at a higher power to attempt to appear closer than it is, it will appear closer relative to all the anchors which receive the signal at a higher power. Such attacks can be detected by consensus among the anchors; while an individual anchor may be fooled on its own, by checking with anchors, the inconsistency is detected. More specifically, when the anchors exchange their distance from the node based on its localization transmission, localization will not be feasible, and the attack is defeated as explained below.

Consider that the actual location of M is (x_m, y_m). Also, assume that nearby anchors are located at (x_1, y_1), (x_2, y_2) ... (x_i, y_i) with respective distance from the localizing node of d_1, d_2, ... d_i. When transmitting at a different power, the detected distances from the anchors will be $d_1 + f$, $d_2 + f$, ... $d_i + f$. This is the key property that makes detecting broadcast manipulation possible. When considering any two anchor nodes, the intersection of the circles with radii $d_j + f$ and $d_k + f$ around the nodes will result in at most 2 candidate location points: possibly 1 candidate point if the circles touch, or 0 if they do not intersect. If there are no candidate points, allowing for localization tolerances, an inconsistency is detected. The set of candidate points is further thinned

when considering other anchors. It is impossible for the candidate location points (if any) to lie on the circle of radius $d_l + f$ around a third anchor l. Informally, this is because starting from the actual location point, it is impossible to find another point simultaneously closer (or farther away) by the same distance from three or more different points. A more formal proof of the above statement can be found in the expanded version of this paper [1]. The inconsistency is easiest to detect if the anchors are not close to each other, to be able to distinguish intentional falsification from localization errors.

In a TDOA localization scheme (transmission of an RF and an ultrasonic pulse concurrently, and measuring the time difference in their arrival to estimate distance), a node may attempt to cheat by sending the RF and ultrasonic pulses at different times. Again, the attack can be detected by consensus since the distance estimated by the separation between the two pulses will be observed to be inconsistent when considered from the different vantage points of the anchors nodes. Thus, this threat can be eliminated. We do not believe this attack to be possible with AoA as a node cannot camouflage its angle to another node, but perhaps such manipulation is possible with smart antennas.

4.2.2 Multiple Unicast Packet Attack

Another attack attempts to prevent consensus between the localization nodes by fooling them with different transmissions. For example, a misbehaving node can directionally send packets to the different localization nodes (for example, each with a different transmission power). The directional transmission can be focused to be received by one or more beacons but not the others. There are two versions of this attack: sequential, and concurrent. In the sequential version, the different localization packets are sent directionally to the different localization nodes one at a time. This attack can be prevented, by having the localization nodes be synchronized with a tolerance of the beacon packet length. This allows the localization nodes to detect the clock skew in the packets fingering them for being different packets rather than one broadcast packet. The concurrent version proceeds by transmitting concurrently from multiple sending radios to the different localization nodes directionally. While this attack cannot be detected by clock skew, MAC level authentication can be used to detect that the transmissions emanated from different radios.

4.2.3 Mobility Attack

Another attack proceeds by obtaining a valid location certificate and then moving to a new location; the validated location information is no longer correct. This attack is a challenge to location verification in general and cannot be easily prevented (at the time of the location verification, the location is correct). The effect of the attack can be minimized by requesting fresh certificates periodically when dealing with a new neighbor. The drawback here is the overhead of localization. Alternatively, trusted nodes,s such as localization nodes, can sample non-trusted nodes transmissions and estimate their distance from it. Reconciling this estimated distance with the node's claimed location from multiple vantage points can provide dynamic detection of mobility attacks.

Another related attack line proceeds by having one node obtain a verified location, and pass it to another node to

use via a secure back channel. This attack can be defended via authentication or the mobility defenses outlined in this section.

4.2.4 Subversion of Localization Nodes

The scheme described so far relies on trust in the localization nodes. Protection against Byzantine failure of localization nodes cannot be directly provided by classical quorum algorithms available in distributed systems [13]. The reason is that the sought consistency is not on a single value of a variable as with classical quorum algorithms; rather, it is on the location of the node as measured by different localization points.

Byzantine attacks can be tolerated by having additional localization points. The presence of inconsistencies in the detected node location can be indicative of two cases: a broadcast manipulation attack was attempted, or one or more of the localization points has an error or was compromised. In this case, we assume that the majority of the localization nodes have not been compromised. The maximum match set is the largest set of localization nodes whose estimate of location is consistent. If the size of this set is less than three, we assume that no valid localization occurred. However, if the size is larger than 3, we assume that the majority is not compromised, and that their estimate is correct because of the consensus of 3 trusted nodes prevents the broadcast manipulation attack.

5. SECURE ROUTING

In this section, we provide a high level discussion of our resilient multi-path geographic routing protocol and trust management scheme as an initial step towards secure geographic routing.

5.1 Resilient Geographic Routing

Although location verification can prevent a Sybil attack in which an adversary claims several geographic locations, a compromised or malicious node can still selectively forward packets disrupting the routing. To address the problem, we propose a probabilistic multi-path routing protocol that is resilient to packet losses due to either an error or malice.

In order to ensure resilient geographic forwarding, we need to ensure that intermediate nodes are living up to their responsibilities and are actually forwarding packets that they are tasked with forwarding (or perhaps provide feedback indicating the reason). Consider nodes A, B, C and D that form a route from A to D. It is difficult for A to verify that B is actually forwarding its packets towards D: C is not in range with it and cannot provide feedback. Moreover, due to the localized nature of interactions, A does not know what nodes are in B's forwarding set.

One check that A can perform is to verify that B at least forwarded the packet to someone via overhearing. When node A sends a packet, it waits for the acknowledgment (ACK) from B, while overhearing B to observe whether or not B forwards the packet.

We note that overhearing test is foolproof for two primary reasons

- A may miss the retransmission due to a collision with another packet

- B may forward the packet, but to a node in the wrong direction or even to a non-existing node. Since A does

not have knowledge of B's neighbors, it is not able to determine that B is not correctly forwarding the packet.

To address the first case, i.e., when a collision occurs at A, a node needs to monitor a neighbor's behavior for multiple packets before evaluating its trustworthiness. Optionally, we may share trust information, for example by allowing highly trusted nodes to periodically exchange the reputation about their neighbors in a cryptographically secure manner. In this way, an individual node can get more trustworthy information about its neighbors derived from the relatively global information. 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.

To address the second case, a mechanism to check whether B is correctly forwarding packets is needed. There are a number of options available here. One option is to query an anchor about the location of the destination B is forwarding a packet to in order to determine that it exists and that it is closer to the destination.

Two key features of our solution are: (1) the use of multi-path routing to increase the probability of using uncompromised paths; and (2) explicit trust management to identify misbehaving nodes and avoid paths that use them. The pseudo code of our protocol is as follows.

1. When a source s wants to transmit a packet toward a destination d for the first time, it establish a shared secret with a local anchor and queries the anchor to get the verified geographic information of the neighbors in a certain range, e.g., twice its radio range. 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 will verify the authenticity and integrity of the received packet using the public key of the sender 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 where the forwarding set FS 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.) Given $\{P_1, P_2, ..., P_N\}$, s independently selects k neighbors to which it will forward the packet where k is the required level of redundancy. We use the roulette wheel selection technique [5] for node selection, 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 ACKs from the neighbors. If s overhears a neighbor forward a packet it checks whether the packet has been forwarded to a legitimate location by referring to its cached location information or querying the anchor, if necessary, to get the related location information. This verification can be performed periodically to reduce its overhead. According to the result, it also adjusts the trust level of the neighbor (described in Section 4.2).

7. If s finds a node i whose trust level $T_i \geq \theta_2$ where $\theta_1 < \theta_2$, it periodically exchange 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 toward d.

5.2 Trust Management

The basic idea of our trust management scheme is to favor well behaving 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. Once a node lies about its location, it is immediately excluded from the FS. Thus, 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 toward the destination will stay longer in the FS to support secure geographic routing.

Once a node constructs a routing table as discussed in Section 4.1, it monitors the behavior of the one hop neighbors to which it forwards packets. (A routing table can also be extended when a sensor node is added to the neighborhood and its location is verified.) Although there could be many possible alternatives, 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 toward d, it will increase the trust level of node i:

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

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 toward d, and thereby, confirming the trustworthiness of the ACK that it receives. Specifically, when a node i is suspected to disrupt routing, showing Byzantine behaviors as discussed before, its trust level is decreased:

$$T_{i_{new}} = \begin{cases} T_i - \Delta t & \text{if } T_i - \Delta t > 0; \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

where Δt is the predefined penalty for each suspicious behavior. Further, via periodic trust information exchange 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.

5.3 Security Analysis and Trade-Offs

By authenticating and encrypting messages, we can prevent an external adversary without cryptographic keys to impersonate a legitimate node or decrypt the cipher-text. Further, it cannot modify data in transit without being detected. Therefore, an adversary is forced to rely on brute force attacks to derive the associated private key from a public key. Or, it has to physically capture sensor nodes and extract the keys.

Unfortunately, we cannot prevent an adversary, i.e., a compromised or malicious node, to disrupt the geographic routing as discussed before. For these reasons, a node overhears one hop neighbors to adjust the trust level of each one hop neighbor. In addition, our 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 exchange.

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 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 trade-off for every application, but one has to select an appropriate value, for example, by using a higher threshold in a more hostile environment.

One may argue that the overhearing process (essential for trust management in our approach) can be energy consuming. We claim that a node can naturally overhear a one hop neighbors radio communication. Therefore, our approach may not significantly increase the energy consumption. In addition, to reduce the energy consumption, a node may randomly choose to overhear a neighbor or not when the energy level becomes low. In the mean time, it may have collected stable information about the neighbors' trustworthiness.

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. In this paper, we provide these design choices rather than claiming that one approach is the best solution to meet the requirements of all possible sensor network applications. In the future, we will thoroughly analyze these trade-off issues and their security implications.

5.4 Virtual Coordinate based Geographical Forwarding

Recently, Virtual Coordinate (VC) systems have been proposed as an alternative to geographic location in GF [4, 19, 15]. VC overlays virtual coordinates on top of the nodes and uses them to route packets towards destinations. Relative to GF, VC removes the need for localization and minimizes the impact of localization errors. However, this comes at the cost of building the virtual coordinates for the nodes.

A typical VC system [4] is initialized according to the information exchanged by all the nodes in the network. Some nodes are chosen as the zero points for each axis. According to the routing packets exchanged between nodes, the virtual coordinates of a node are set by incrementing the smallest values among all its neighbors. Usually for a 2-dimensional plane, a 3-dimensional VC system is required.

Our simulation results show that the behavior of GF based on VC is nearly the same as the one based on actual location. Besides the potential threats common to GF, there are some new threats introduced when using VC. Here we mention two of the threats unique to VC based GF; more careful consideration of the security of these protocols is left to future work.

- VC Initialization attacks. Since most VC systems are set up based on the information exchanged between neighbors, a malicious node may attack the whole system by spreading wrong information to its neighbors. This could cause the initialization of the virtual coordinates to fail, or result in VCs with loops or suboptimal paths. This attack is similar to the location verification problem. However, since virtual coordinates are used instead of geographical location, VC verification is a different and more difficult problem.

- Alias attacks. For some VC systems, virtual coordinates may be shared among several nodes. A malicious node may locate itself near the destination to intercept information by setting its virtual coordinates location the same as the destination and pretending the node id as that of the destination. The location verification would not help resolve this problem since the virtual coordinates of the malicious node are correct.

6. RELATED WORK

Wireless sensor networks are exposed to numerous security attacks. Karlof et al. [9] 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. 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, 8] exist.

Localization has been well studied [17, 23, 24, 2, 6, 14, 3, 16]; however, most existing approaches do not address

security. Sastry et al. [22] 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. Our verification work presents a different approach to location verification that does not require an ultrasonic channel and provides more accurate verification (full location verification, rather than distance to a verifier). In addition, it introduces and addresses some attacks that were not considered by them.

Lazos et al. [11] 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 is concerned with attacks on the localization mechanism to cause nodes to have erroneous location information, but 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) [10] and Geographic and Energy Aware Routing (GEAR) [25] 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. 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 [21] 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, subverts our secure geographic routing protocol. In fact, our protocol can cooperate with a non-secure geographic routing protocol by supporting location verification and online detection of and tolerance against Sybil and selective forwarding attacks.

ARRIVE [7] 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. Further, we correlate the ACK and overhearing unlike ARRIVE, while allowing trustworthy nodes to exchange the trust information between them to derive a more global view when the environment is considered relatively benign.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we outlined some of the security threats that arise in the context of Geographical Forwarding (GF). While

security in multi-hop wireless networks such as WSNs and Ad hoc networks has been well studied, these have mostly focused on traditional routing protocols. The nature of GF makes it vulnerable to a different set of attacks and require specialized solutions for securing them.

In this paper, we studied two areas of vulnerabilities in GF. First, GF trusts nodes to supply their location information and uses it in determining forwarding decisions. There is no protection against nodes falsifying this information. We presented an approach for secure and validated localization. The key idea in the approach is to task more trusted anchors with determing sensor node locations. This prevents nodes from fabricating location information. However, a number of possible attacks remain. We discussed these attacks and outlined solutions to them.

The second area of vulnerability we considered is routing: even if location information is accurate, nodes may still misbehave, for example by dropping or manipulating packets. The proposed solution here is to use probabilisitic multi-path, with trust-based route selection, to dynamically avoid untrusted paths and continue to route packets in the presence of attacks. We discussed the proposed approach in detail, outlining alternative choices. We also considered possible attacks and defenses against them.

There are a number of open research issues that remain to be addressed. First, we did not consider the security implications of voids. Voids are typically bypassed by a secondary routing mechanism called face routing [10]. We did not consider possible attacks on this component of GF algorithms. We also did not pursue the implications of using proximity based localization algorithms [3]. Finally, virtual coordinate routing is a promising routing approach that virtualizes node location. We believe that virtual coordinate routing is vulnerable to different attacks than GF and merits further study to identify and address them.

8. REFERENCES

[1] N. Abu-Ghazaleh, K. Kang, and K. Liu. Towards resilient geographic forwarding in wireless sensor networks. Technical report, Binghamton University, CS. Dept., 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] D. Goldberg. *Genetic Algorithm in Search, Optimization and Machine Learning.* Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1989.

[6] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher. Range-Free Localization Schemes for Large Scale Sensor Networks. In *MobiCom'03*, 2003.

[7] 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.

[8] C. Karlof, N. Sastry, and D. Wagner. TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. In *ACM SenSys*, 2004.

[9] C. Karlof and D. Wagner. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. In *Sensor Network Protocols and Applications*, 2003.

[10] B. Karp and H. Kung. GPSR: Greedy Perimeter stateless Routing for Wireless Networks. In *MobiCom*, 2000.

[11] L. Lazos and R. Poovendran. SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks. In *the ACM workshop on Wireless security*, 2003.

[12] 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.

[13] D. Malkhi and M. Reiter. Byzantine quorum systems. In *Proc. ACM Symposium on Theory of Computing*, pages 569–578, 1997.

[14] 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.

[15] D. Nicol, M. Gloldsby, and M. Johnson. Simulation analysis of virtual geographic routing. In *Proc. of the 2004 Winter Simulation Conference*, 2004.

[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] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *MobiCom'03*, 2003.

[20] S. Ratnasamy, D. Estrin, R. Govindan, B. Karp, S. Shenker, L. Yin, and F. Yu. Data-centric storage in sensornets. In *Proceedings of the First ACM SIGCOMM Workshop on Hot Topics in Networks*, Oct. 2002.

[21] 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.

[22] N. Sastry, U. Shankar, and D. Wagner. Secure Verification of Location Claims. In *the ACM workshop on Wireless security*, 2003.

[23] A. Savvides, C. Han, and M. Srivastava. Dynamic Fine Grained Localization in Ad-Hoc Sensor Networks. In *Mobicom*, 2001.

[24] S. Čapkun, M. Hamdi, and J.-P. Hubaux. GPS-free positioning in mobile ad-hoc networks. In *HICSS*, 2001.

[25] 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.