Available online at www.sciencedirect.com

ScienceDirect

J. Parallel Distrib. Comput. 67 (2007) 215–228

Journal of Parallel and Distributed Computing

www.elsevier.com/locate/jpdc

Location verification and trust management for resilient geographic routing[☆]

Ke Liu, Nael Abu-Ghazaleh*, Kyoung-Don Kang

CS Department, Binghamton University, USA

Received 6 March 2006; received in revised form 29 July 2006; accepted 3 August 2006
Available online 24 October 2006

Abstract

In this paper, we consider the security of geographic routing (GR) protocols. In GR, neighbors exchange their location information. Based on this information, a node forwards packets to the neighbor that is closest to the destination. Although GR is widely used in ad hoc and wireless sensor networks, its security has rarely been studied; there are a number of attacks that are possible on GR. In one attack, misbehaving nodes can falsify their location information. Also, malicious nodes can drop packets that they need to forward towards the destination. The first contribution of the paper is to propose a location verification algorithm to address the attacks falsifying the location information. The second contribution of the paper is to propose approaches for trust-based multi-path routing, aiming to defeat attacks on GR. We discuss the proposed approaches in detail, outlining possible attacks and defenses against them. In addition, we show, via simulation, how trust-based route selection is able to circumvent attackers and route around them. This paper summarizes and extends results reported by the authors in a previous article [K.-D.K. Nael, B. Abu-Ghazaleh, K. Liu, Towards resilient routing in WSNs, in: Proceedings of the First IEEE/ACM Workshop on QoS and Security in Wireless Networks (Q2SWinet 2005), 2005, pp. 71–78].

Keywords: Geographic routing; Location verification; Trust management

1. Introduction

Wireless sensor networks (WSNs) hold the promise of revolutionizing sensing across a large number of disciplines. Geographic routing (GR) protocols, such as GFG [2], greedy perimeter stateless routing (GPSR) [20,21] is an attractive approach for routing in WSNs because of its low overhead and localized interactions. In GR, nodes only interact with their neighbors making localized forwarding decisions. More specifically, nodes exchange the location information with their neighbors. Based on location information, a node forwards a packet to a neighbor that is closer to the destination.

While a number of studies consider security in multi-hop wireless networks such as WSNs and ad hoc networks, they have mainly focused on conventional routing protocols. Very little work has studied the issue of securing GR and the underlying localization mechanisms. There are fundamental differences between GR and traditional routing algorithms. GR relies on an underlying localization mechanism that allows the sensor nodes to determine their geographic locations [6,15,32]. Further, the use of locations, rather than node IDs, introduces additional security concerns. For these reasons, there are a number of unique attacks that are possible on GR. In this paper, we outline some of these attacks and present solutions to them.

In a class of possible attacks on GR, misbehaving nodes can falsify their location information, because there is no oversight mechanism verifying that a node is actually at the position it claims. An attacker can exploit this vulnerability to force sensor nodes to use suboptimal routes or misroute packets, for example, by launching Sybil attacks [19]. The first contribution of the paper is to propose a location verification algorithm that adresses this problem. The proposed approach has several advantages over another recently proposed location verification algorithm [38]. (A detailed discussion is given in Section 4.)

[☆] This work was supported in part by NSF Grants CNS-0614771 and CNS-0454298 as well as US Army project W911SR-05-C-0014.

* Corresponding author.
E-mail addresses: kliu@cs.binghamton.edu (K. Liu), nael@cs.binghamton.edu (N. Abu-Ghazaleh), kang@cs.binghamton.edu (K.-D. Kang).

0743-7315/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2006.08.001

Location verification allows nodes to have confidence in the location of their neighbors, preventing many straightforward attacks on routing. However, there are other possible attacks on routing. For example, a node may simply drop packets that it has to forward. The second contribution of the paper is to propose approaches for increasing the resiliency of GR to attacks on routing. Specifically, we propose a probabilistic multi-path routing protocol that is able to continue to route packets even in the presence of blackhole or selective forwarding attacks [19] by dynamically avoiding untrusted paths via trust-based route selections. By seamlessly integrating the location verification and trust-based routing, we can prevent attacks based on false location information, while making GR resilient to routing attacks.

The remainder of the paper is organized as follows. In Section 2, GR and localization backgrounds are given. Our threat model is described in Section 3. Our location verification algorithm and possible attacks and defenses are discussed in Section 4. A resilient GR protocol is proposed in Section 5. The security of the protocol and the related trade-offs are also discussed. In Section 6, via a simulation study, the performance of the resilient protocol is compared to a well-known insecure GR protocol [20]. The related work is presented in Section 7. Finally, Section 8 concludes the paper and discusses future work.

2. Background

This section presents background information about GR and localization necessary to explain the security issues in GR.

2.1. Geographic routing

GR typically consists of two parts: geographic forwarding and a complementary routing for void avoidance, called face routing or perimeter routing. Geographic forwarding is a greedy routing algorithm based on geography. For a given node, all its one-hop neighbors closer to the destination belong to the *forwarding set* (FS) for that destination. As shown in Fig. 1(a), the node forwards an incoming data packet to the neighbor that is in the FS and nearest to the destination. GR is attractive, since it only requires nodes to maintain the location of their one-hop neighbors. Also, routing decisions can be made locally and dynamically as discussed before.

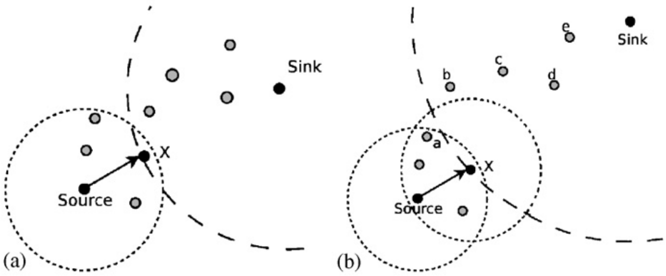

Fig. 1. Geographic routing example: (a) X is source's nearest neighbor to sink; (b) voids: X is a local minima.

GR does not always succeed in the greedy phase described above. When the forwarding node, e.g., node x in Fig. 1(b), has no one-hop neighbor closer to the sink than itself, it cannot further forward the incoming data packet. As a result, 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 [2,20] or backtracking towards a beacon [12,24]) is used to route around the void, and more optimizations coming out [10,13,21]. Since more and more applications based on GR are proposed [11,16,25,26], the security of GR becomes more important.

2.2. Localization

Sensor nodes are often not equipped with their own global positioning system (GPS) devices. Moreover, in indoors settings, a GPS signal is not available. 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 [32] is the calculation of position information based on the distance estimated from the anchors. A two-dimensional (2D) position requires three distance measurements. In Fig. 2(a), for example, a node can compute its location by using the estimated distances from the three nearby beacons. The distance can be estimated based on, for example, 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 (AoA) information from three known anchors.
- *Proximity-based or range-free localization*: Instead of relying on sophisticated approaches to measure either the distance or AoA 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 [6,15].
- *Other approaches*: Approaches based on the 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 the location. This kind of localization algorithms requires a compiling database which is not available for most sensor network applications. In dead reckoning, the initial location of a mobile sensor is known. Motion sensors, e.g., accelerometers, are used to measure the velocity which is then used to estimate traveled distance, via dead-reckoning, between velocity measurements.

In addition, several schemes [15,32] have been proposed for multi-hop localization, in situations where the number of beacons is insufficient to directly localize all the nodes in the WSN. For example, as shown in Fig. 2(b), the DV-hop algorithm [32]

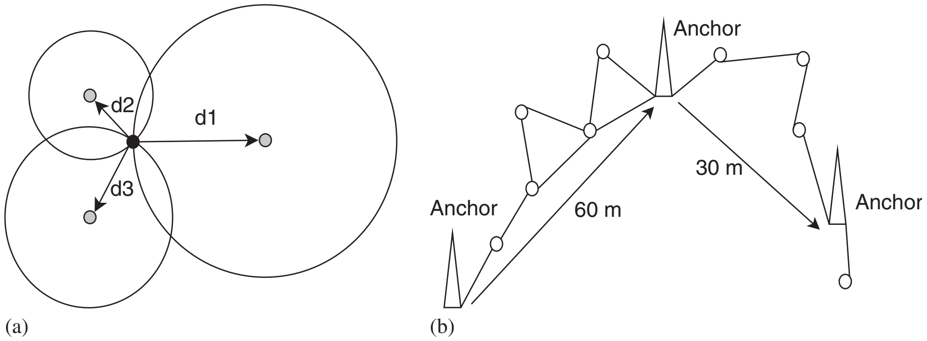

Fig. 2. Localization in sensor networks: (a) triangulation-based localization; (b) multi-hop localization.

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 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 the minimum hop-count distance 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 every node. Individual nodes use the average distance per hop, along with the hop count to known beacons, to calculate their local positions using lateration. In this paper, we focus on the location verification problem based on the triangulation schemes. Securing other localization methods is reserved for future work.

3. Threat model and assumptions

It is impossible to consider all security problems across varying WSN configurations. Therefore, we make the following assumptions:

- There are two types of nodes: anchors and sensor nodes. An anchor is assumed to know its own location, for example, using a GPS. We assume that the anchor density is sufficient to allow most sensor nodes to localize with anchors in one-hop range. We also assume anchors are trusted and they can communicate with each other in a secure manner. We consider the effect of attacks on anchors in Section 4.2.4.
- Although either a secret key or public key system can be used for our approach, we assume an efficient public key system, e.g., *ecTinyos* [27] 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 shared key for more efficient communications, similar to other approaches [18,35]. Therefore, we assume it is possible to support the message confidentiality, integrity, and authenticity using public key or secret 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 GR 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 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, e.g., a battle field.
- Finally, we assume the network is dense enough such that a sensor node has several one-hop neighbors in its radio range. Hence, multi-path routing for resiliency is possible.

4. 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 [40]. To address this unique threat to GR, we propose a location verification algorithm in this section.

4.1. Basic approach

To prevent sensor nodes from falsifying their locations, Sastry et al. have proposed a location verification scheme [38] 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 included in the 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 specific hardware, while verifying the claimed location relative to only one verifier. Furthermore, 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.

In this paper, we propose an alternative approach for location verification. The key idea is to reverse the localization procedure such that a non-trusted sensor node is not allowed to generate its own location estimate. The proposed scheme can be used for triangulation-based localization methods including the ones using the radio signal strength (RSS), time of arrival (ToA), time difference of arrival (TDOA) and AoA discussed in Section 2. In the proposed scheme, localization is initiated by having a sensor transmit a localization request to the anchors in the neighborhood. This request is received by multiple (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. In this way, we can disallow a non-trusted sensor node from spreading false location information. Also, note that our approach can be used to verify a location even if it is not the primary localization algorithm used by the sensors.

4.2. Possible attacks and countermeasures

In the remainder of this section, we overview possible attacks on the proposed location verification 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 the localization by exploiting the underlying localization mechanism. For example, when the RSS is used, the distance is estimated based on the measured signal strength. A node may attempt to cheat by transmitting at a higher/lower power level to appear closer/farther than it is. A similar attack can be attempted to a TDOA-based localization scheme in which a node is required to concurrently transmit an RF and an ultrasonic pulse. A malicious node may attempt to cheat by sending the RF and ultrasonic pulses at different times. When the AoA is used for localization, the broadcast manipulation might be difficult as a node cannot camouflage its angle to another node unless it has smart antennas.

If a misbehaving node M transmits a localization request at a higher/lower power level to appear closer/farther than it is, it will appear closer/farther relative to all the anchors that receive the signal. Such attacks can be detected by deriving a consensus among the anchors. Although a single anchor may be fooled on its own, several anchors can collaborate to detect the inconsistency, if any. When a malicious node tries to falsify its location, in our approach, the collaborating anchors will find the localization infeasible, thereby defeating the attack. Further, our approach can derive the correct location even under a falsifying attack.

Consider that the actual location of M is (x_m, y_m). Also, assume that nearby anchors are located at $(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i)$ with respective distances $d_1, d_2, \ldots, d_i$ to M. If M transmits at higher or lower power, the detected distances between M and the anchors will be $d_1 + f, d_2 + f, \ldots, d_i + f$. 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 two candidate location points, while getting possibly one candidate point if the circles touch, or 0 if they do not intersect. An inconsistency is detected, if there are no candidate points even after considering localization and distance estimation error tolerance.

The set of candidate points is further thinned by considering other anchors. The candidate location points, if any, cannot 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 for a misbehaving node to find another point simultaneously closer (or farther away) by the same distance from three or more different points. In fact, it is possible for the anchors to derive the true location of the node attempting this attack, by solving for f directly. Especially, the inconsistency can be easily detected if the anchors are far enough to distinguish intentional falsification from localization errors.

We analytically prove the correctness of the above approach as follows. In conventional trilateration, given three distance estimates of a node to three known locations, it is possible to estimate M's location. Thus,

$$(x_1 - x_m)^2 + (y_1 - y_m)^2 = d_1^2, \tag{1}$$

$$(x_2 - x_m)^2 + (y_2 - y_m)^2 = d_2^2, \tag{2}$$

$$(x_3 - x_m)^2 + (y_3 - y_m)^2 = d_3^2, \tag{3}$$

can be solved to find the x_m and y_m values in 2D space. (An additional anchors, if available, can also be used.) When the attacker is attempting this attack, all the observed values of $d_1, \ldots, d_i$ will be off by the same distance f. Given three anchors, the equations above will not have a single solution unless the attacker and all the anchors lie on a straight line.

Furthermore, if the anchors assume that this attack is being attempted, they can derive the correct location of the attacker as follows. First, we note that the estimated distance at the anchor, e_i, is actually $(d_i + f)$. Thus, the equations can be updated per anchor as follows:

$$\sqrt{(x_i - x_m)^2 + (y_i - y_m)^2} + f = e_i, \tag{4}$$

which simplifies to

$$(x_i - x_m)^2 + (y_i - y_m)^2 = (e_i - f)^2 = e_i^2 + f^2 - 2e_i f. \tag{5}$$

Thus, we can solve Eqs. (1)–(3) for *f* and then substitute the *f* value to Eq. (5) to obtain the true location.

Consider the following example where the attacker is at (5,5) and the anchors are at (0,10), (10,10) and (10,0). When the node localizes correctly, the observed distance between the sensor node and each of the three anchors will be $5\sqrt{2}$. This results in the following trilateration equations:

$$x_m^2 + 100 - 20y_m + y_m^2 = 50, \tag{6}$$

$$200 - 20x_m + x_m^2 - 20y_m + y_m^2 = 50, \tag{7}$$

$$100 - 20x_m + y_m^2 = 50. \tag{8}$$

This can be simplified to produce the correct answer. For example, subtracting the first equation from the second equation yields

$$100 - 20x_m = 0, \tag{9}$$

which yields $x_m = 5$. Similarly, the third equation can be subtracted from the second to yield $y_m = 5$.

Now consider that the node at (5, 5) transmits at a lower power level to appear two units farther than its actual location from all the anchors. In this case, our approach tries to solve the resulting equations acquired by substituting the anchors' locations and the sensor node's (falsely) claimed location into Eqs. (1)–(3), finding an inconsistency as follows:

$$x_m^2 + 100 - 20y_m + y_m^2 = (5\sqrt{2} + 2)^2 = 82.28, \tag{10}$$

$$200 - 20x_m + x_m^2 - 20y_m + y_m^2 = 82.28, \tag{11}$$

$$100 - 20x_m + y_m^2 = 82.28. \tag{12}$$

In this case, the equations can be solved in the same way as before, yielding the same answer. However, substituting these x_m and y_m values into any of the three equations will quickly reveal the infeasibility of the solution as the left- and right-hand sides of the equation are not equal.

At this stage, the anchors can also derive the value *f* which is the distance by which the node is attempting to lie using Eq. (5) as follows:

$$\begin{aligned} x_m^2 + 100 - 20y_m + y_m^2 &= (e_i - f)^2 \\ &= 82.28 - 18.11f + f^2, \end{aligned} \tag{13}$$

$$\begin{aligned} &200 - 20x_m + x_m^2 - 20y_m + y_m^2 \\ &\quad = 82.28 - 18.11f + f^2, \end{aligned} \tag{14}$$

$$\begin{aligned} &100 - 20x_m + y_m^2 \\ &\quad = 82.28 - 18.11f + f^2. \end{aligned} \tag{15}$$

Thus, we can solve the above equations for x_m and y_m. In this case, we get two roots for *f*: 2 and 16.1. Note that the 16.1 root is not feasible since all the three of d_1, d_2 and d_3 would become negative. Thus, localization is possible even under the broadcast manipulation attack.

4.2.2. Multiple unicast packet attack

A possible attack may attempt to prevent consensus between the anchors by fooling them with different unicast transmissions. For example, a malicious node can directionally send requests to different anchor nodes possibly using a different level of transmission power for each request. The directional transmission can be engineered 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 synchronizing the localization anchors with the tolerance of a beacon packet length. The localization nodes can detect the clock skew existing in the serial attack. The concurrent version proceeds by concurrently and directionally transmitting multiple requests, using multiple radios, to different localization nodes. While this attack cannot be detected based on the clock skew, MAC level authentication can be used to detect that the transmissions emanated from different radios.

4.2.3. Mobility attack

In this attack, a malicious node can obtain a valid location certificate and then move to a new location. Consequently, the validated location information is no longer correct. This attack cannot be easily prevented, because the location is correct at the time of the location verification. The effect of the attack can be minimized by requiring sensor nodes to have anchors periodically renew their certificates at the cost of the increased overhead for localization. Alternatively, trusted nodes such as localization nodes can sample non-trusted nodes' transmissions and estimate their distance to themselves. Reconciling this estimated distance with the node's claimed location from multiple vantage points can provide dynamic detection of mobility attacks. In another related attack, a node may obtain a verified location and pass it to another node through a back channel. This attack can be defended via authentication or the other defense mechanisms outlined above.

4.2.4. Subversion of localization nodes

The scheme described so far assumes that the localizing anchor nodes are trusted and not compromised. Protection against Byzantine failures of localization nodes cannot be directly provided by classic quorum algorithms in distributed systems [28]. The reason is that the sought consistency is not on a single value of a variable considered in traditional quorum algorithms, but it is on the location of the node measured by different localization points. Byzantine attacks can be tolerated by having additional localization points. The presence of detected inconsistencies in the node location may indicate two possibilities. First, a broadcast manipulation attack has been attempted. Second, at least one anchor has an error or it has been compromised. To handle this case, we assume that the majority of the localization nodes have not been compromised. If the size of the maximum match set—the largest set of localization nodes whose location estimates are consistent—is smaller than three, we assume that no valid localization has occurred. In contrast, if the size is larger than three, we consider their estimate is correct, because the consensus of three trusted nodes can prevent the broadcast manipulation attack as discussed before.

5. Secure routing

In this section, we propose a resilient multi-path GR protocol and trust management scheme as an initial step towards secure GR.

5.1. Resilient GR

Although location verification can prevent an attack predicated on falsifying location information a compromised or malicious node can still selectively forward packets disrupting the routing. To address this problem, we propose a probabilistic multi-path routing protocol that is resilient to packet losses due to either an error or a malice.

In order to ensure resilient geographic forwarding, 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). Consider nodes A–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. If C is not within A's range, C cannot provide a feedback. Moreover, due to the localized nature of interactions in GR, A does not know what nodes are in B's FS.

What is needed is a *forwarding verification scheme*. One such approach 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 the overhearing test is not a foolproof check for two primary reasons:

- Node A may miss the retransmission due to a collision with another packet.
- 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 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. To address the second case, a mechanism to check whether B's is forwarding packets to a correct next hop 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. We note that this scheme can be defeated by two attackers in sequence and end-to-end checks are needed for stronger verification of forwarding behavior. We do not pursue this difficult problem further; it deserves separate treatment which we leave to future work. Instead, we assume that a forwarding verification check exists and use it to adapt the trust level; for simplicity, we assume that the overhearing test works as a forwarding verification check.

Optionally, we may share trust information that can be built more quickly and accurately 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 wider perspective 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.

Two key features of our solution are: (1) the use of multipath 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 pseudocode 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 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 request to send (RTS) 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 clear to send (CTS) 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 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 5.2.) Given $\{P_1, P_2, \ldots, P_N\}$, 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 [14] 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 (alternative forwarding verification checks may be employed). This verification can be performed periodically to reduce its overhead. According to the results, it also adjusts the trust level of the neighbor.

7. If s finds a node i whose trust level $T_i \geqslant \theta_2$ 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.

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 towards the destination will stay longer in the FS to support secure GR.

Once a node constructs a routing table as discussed in Section 5.1, 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 towards d, it will increase the trust level of node i:

$$T_{i_{\text{new}}} = \begin{cases} T_i + \delta t & \text{if } T_i + \delta t \leqslant 1, \\ 1 & \text{otherwise,} \end{cases} \tag{16}$$

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, its trust level is decreased:

$$T_{i_{\text{new}}} = \begin{cases} T_i - \Delta t & \text{if } T_i - \Delta t > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

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 GR, 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 ciphertext. Further, the adversary 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.

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 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 tradeoff 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 forwarding verification process (essential for trust management in our approach) can be energy consuming. To reduce the energy consumption, a node may randomly invoke the verification check on a neighbor or not when the energy level becomes low. In the meantime, 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, $\Delta t \leqslant \delta t$ 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.

5.4. *GR based on virtual coordinates*

Recently, virtual coordinate (VC) systems have been proposed as an alternative to geographic location in GR [31,7,12,9,36,24]. VC overlays VCs of the nodes and uses them to route packets towards destinations. Relative to GR, VC removes the need for localization and minimizes the impact of localization errors. However, this comes at the cost of building the VCs for the nodes.

A typical VC system [9] 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 VCs of a node are set by incrementing the smallest values among all its neighbors. Usually for a 2D plane, a three-dimensional VC system is required.

Our simulation results show that the behavior of GR based on VC is nearly the same as the one based on the actual location information. Besides the potential threats common to GR, there are some new threats introduced when using VC. Here, we mention two of the threats unique to VC-based GR; more careful consideration of the security of these protocols is reserved for 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 VCs to fail, or result in VCs with loops or suboptimal paths. This attack is similar to the location verification problem. However, since VCs are used instead of geographical location, VC verification is a different and more difficult problem.
- *Alias attacks*: For some VC systems, VCs may be shared among several nodes. A malicious node may locate itself near the destination to intercept information by setting its VC location the same as the destination and use the destination's node ID. The location verification would not help resolve this problem since the VCs of the malicious node are correct.

6. Experimental evaluation

We have implemented our design of resilient GR with trust management in the network simulator NS2 version 2.29 [41]. [1] The details of the experimental settings and results are given next.

6.1. *Design of experiments*

The trust management scheme is implemented by extending the GPSR algorithm [20]. We also extended the IEEE 802.11 to support overhearing needed in our approach. The extended version is called resilient geographic routing (RGR), while the original GPSR is called insecure GPSR (InS GPSR) in the remainder of this section.

Table 1
Simulation parameters

Radio range	30 m
Bandwidth	2 Mbps
Data payload	64 B
Packet size	158 B
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 W
Receiving power	0.2 W

In our experiments, 100 sensors are regularly deployed in 10×10 grids covering an area of $200 \times 200\,\mathrm{m}^2$, in which each node is located at 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 Fig. 3. In scenario 1, only one attacker resides on the shortest path from the source to destination constructed by GPSR. In scenario 2, all the nodes constructing the shortest path are attackers. In scenario 3, the nine attackers form a wall across the network and try to separate the source and destination. For scenarios 1–3, we fix δt as 0.01.

Scenarios 4 and 5 share the same network structure. In Section 6.2, scenarios 4 and 5 use the different δt values, i.e., 0.01 and 0.02, respectively. In Section 6.3, 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.

6.2. *Basic study*

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

6.2.1. *Delivery ratio*

Fig. 4(a) shows the delivery ratio achieved in the different scenarios. In scenarios 1–3, InS GPSR completely fails if any attackers are on the path from the source to destination. As a result, no packet is received by destination. RGR's high delivery ratio in 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 Fig. 4(a), the delivery ratio of RGR in 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 scenario 5, we have performed the same

[1] The source code for the experiments used in this paper is available from the following website http://www.cs.binghamton.edu/~kliu/publication.html.

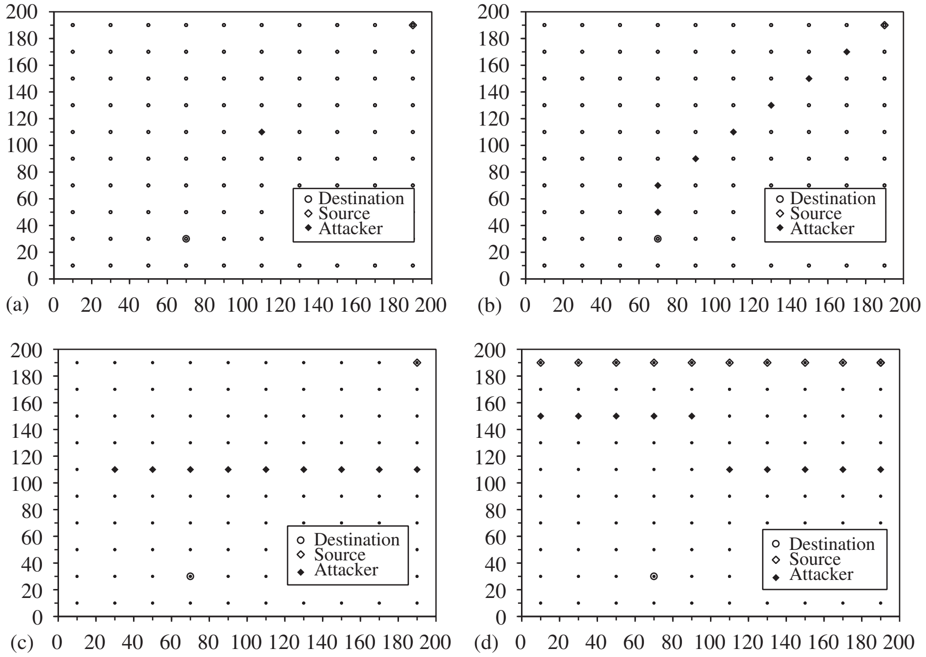

Fig. 3. Scenarios for the simulation study: (a) scenario 1; (b) scenario 2; (c) scenario 3; (d) scenarios 4 and 5.

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. Thus, our trust management is not intrusive when it is not required.

6.2.2. Path stretch

The path length is shown in Fig. 4(b). In scenarios 1–3, 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 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 Fig. 4(b), the path lengths observed in 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 a little 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, the path stretch observed in RGR is stable in the same network structure under the same attacking mode.

Figs. 4(c) and (d) show the total energy and per packet energy consumptions. With no attack, the energy consumption of RGR is similar to InS GPSR. Under an attack, RGR consumes more energy than InS GPSR, but it achieves the higher delivery ratio as discussed before. As shown in Fig. 4(c), the total energy consumption of RGR in scenario 5 is higher than that in scenario 4 due to its higher delivery ratio than that in scenario 4 as shown in Fig. 4(a). However, as shown in Fig. 4(d), RGR consumes less energy per packet in scenario 5 than in scenario 4 because of the larger δt value that can better keep effective paths.

6.3. Effect of trust adjustment parameters

In Section 6.2, adjusting the value of δt has resulted in the considerably different performance of RGR in scenarios 4 and 5. Thus, in this section, we analyze the relationship between δt and ΔT and the impact.

6.3.1. 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 $\Delta t = 0.1$. Fig. 5(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. 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

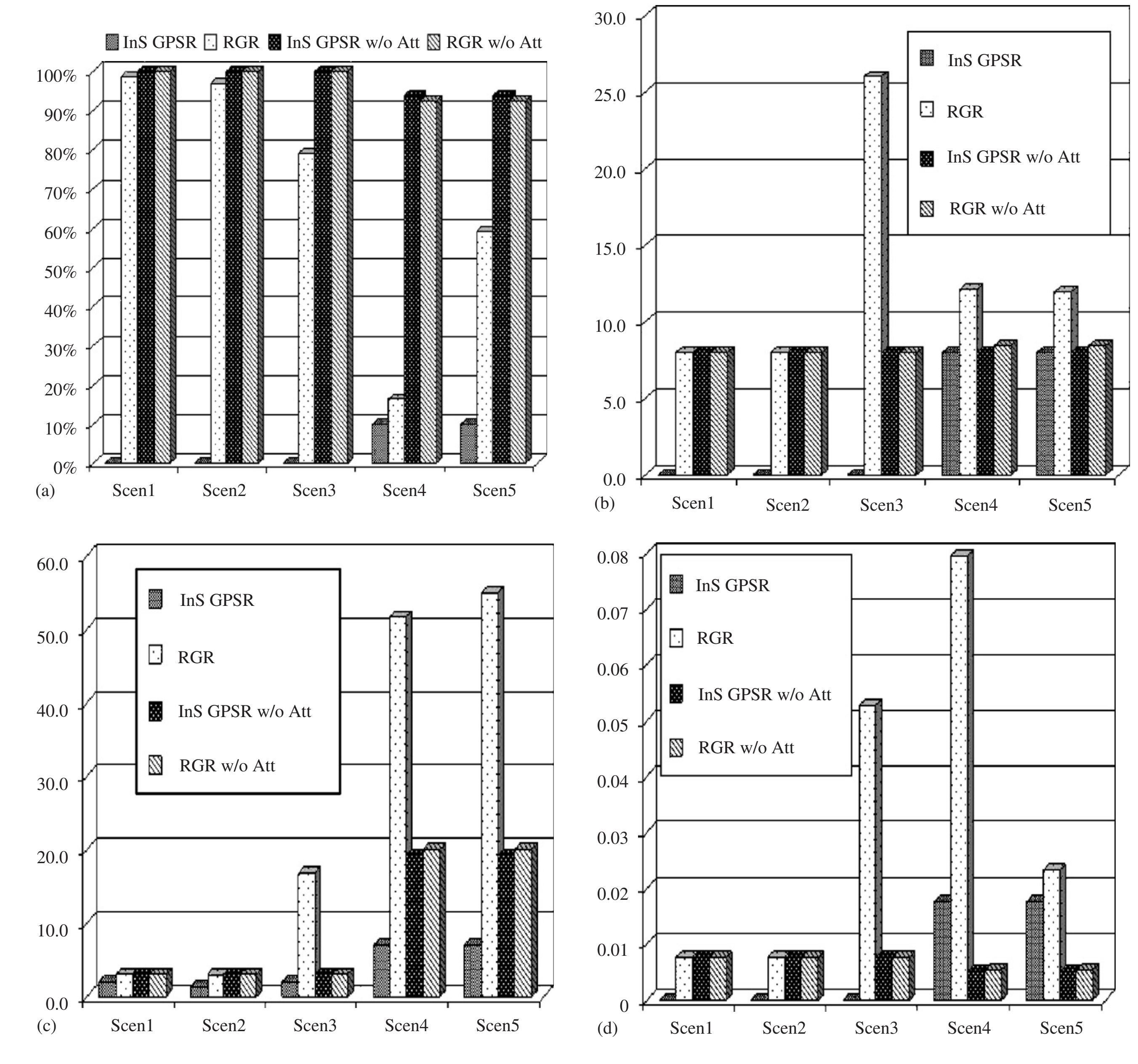

Fig. 4. Basic study: (a) delivery ratio; (b) path length of the received packets; (c) total energy consumption; (d) energy consumption per received packet.

congestion or collision-related packet losses. In addition, Figs. 5(b) and (c) show that a larger δt is more energy conserving, since more packets go through some hop-spot nodes on shorter paths, decreasing the energy consumption.

6.3.2. *Impact of traffic*

In this subsection, we study the impact of the traffic load on the performance of RGR. In a network structure and attacking mode of scenario 5, we fix $\Delta T = 0.1$ 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.

Fig. 6(a) 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 similar behavior can be observed in Fig. 6(b). With fewer data sources, a lower data rate may sacrifice a larger portion of packets for training, consuming more energy for each received packet. As the number of the sources increases, the per-packet energy consumptions for the different data rates become similar.

7. Related work

WSNs are exposed to numerous security attacks. Karlof and Wagner [19] discuss the possible routing disruption attacks and countermeasures. They pointed out that false location claims

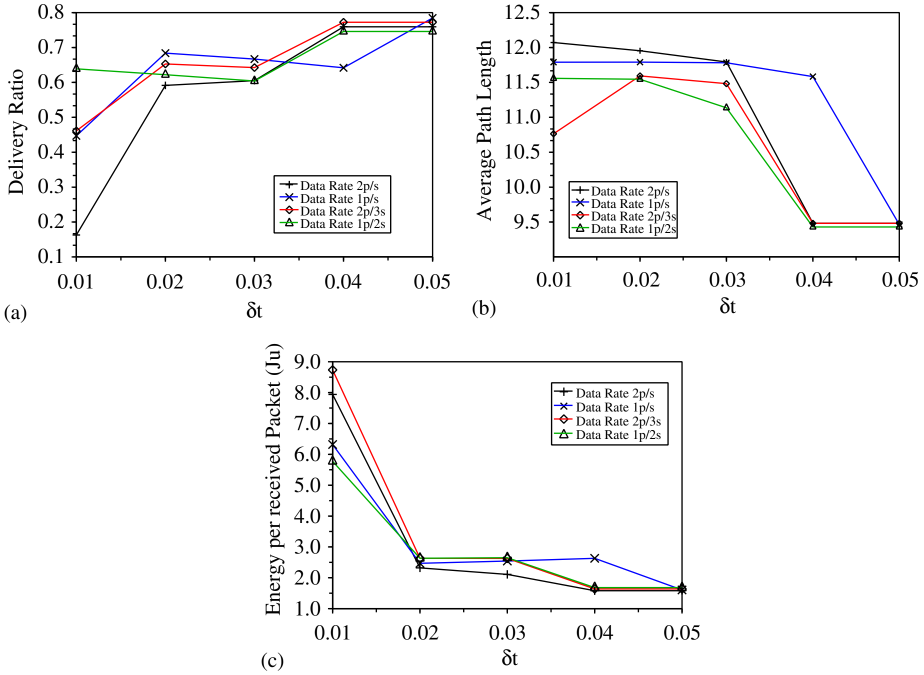

Fig. 5. Impact of δt: (a) delivery ratio; (b) path length; (c) energy consumption.

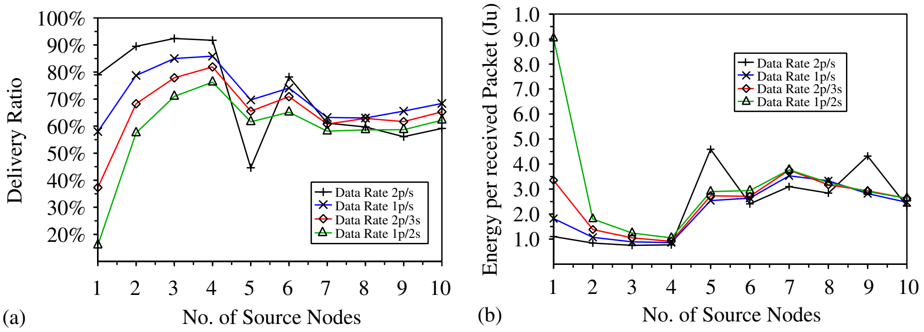

Fig. 6. Traffic impact: (a) traffic impact: delivery ratio; (b) traffic impact: energy consumption.

can seriously disrupt a GR protocol, while suggesting multipath routing as a countermeasure against selective forwarding attacks. 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 [18,35].

Localization has been well studied [1,6,8,15,30,32,33,39]; however, most existing approaches do not address security. Sastry et al. [38] 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 present a different approach 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 [38]. In addition, we have identified and addressed new attacks related to location verification. Lazos and Poovendran [22] 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 it does not prevent a misbehaving node from providing false estimates of its own location to its neighbors.

GR protocols such as GPSR [20] and geographic and energy aware routing (GEAR) [43] 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 GR 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 [37] 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 GR 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 GR protocol. In fact, our protocol can cooperate with a non-secure GR protocol by supporting location verification and online detection of and tolerance against Sybil and blackhole/selective forwarding attacks.

Intrusion detection and secure routing in wireless networks have recently attracted a lot of attention [3–5,42]. A novel intrusion detection scheme [5] has been proposed to detect frauds and immediately notify the users via neural network classification techniques. Our current work seeks to enable a node to estimate the trustworthiness of neighboring nodes via a lightweight approach considering the stringent resource constraints in WSNs. Lakshmi et al. [42] extend AODV [34] for not only external attack prevention but also internal attack detection and correction. Their approach can significantly reduce the routing load as the percentage of the compromised nodes increases compared to the baseline approach without the attack detection and correction scheme. At the same time, the packet delivery ratio of their approach is similar to the baseline. Along the line, our work can be extended in the future to support more resilience against compromised beacons and sensor nodes. Boukerche et al. [3] propose an efficient anonymous routing protocol for wireless ad hoc networks. It allows trustworthy intermediate nodes to take a part in the path construction, while supporting the anonymity of the communicating nodes to avoid a possible traffic analysis. Trust and reputation management is key to security in ad hoc networks; Boukerche and Li [4] propose a trust and reputation management scheme in WSNs considering the system perspectives to reduce the energy, bandwidth consumption, and delay for trust and reputation management. We plan to adapt these techniques to extend our trust management scheme.

ARRIVE [17] is a robust routing protocol applicable to WSNs 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 GR 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. Several distributed algorithms [23] extend existing GR protocols by supporting location-based multi-path routing. This work is complementary to our work in that we focus on the security aspect, while it focuses on the network performance.

8. Conclusions and future work

In this paper, we outlined some of the security threats that arise in the context of geographical forwarding (GR). While security in multi-hop wireless networks such as WSNs and ad hoc networks has been well studied, most existing work has focused on traditional routing protocols. The nature of GR makes it vulnerable to a different set of attacks and therefore requires specialized solutions for securing them.

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

Another key contribution is RGR. Even if location information is accurate, nodes may still misbehave, for example, by dropping or manipulating packets. To dynamically avoid untrusted paths and continue to route packets even in the presence of attacks, the proposed solution uses 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 RGR protocol to a well-known GR protocol.

There are a number of open research issues that remain to be addressed. First, the general problem of trust management in wireless ad hoc networks remains an open problem. Second, we did not consider the security implications of voids. Voids are typically bypassed by a secondary routing mechanism called face routing [20]. We did not consider possible attacks on this component of GR algorithms. We also did not pursue the implications of using proximity-based localization algorithms [6]. Finally, virtual coordinate (VC) based routing is a promising routing approach that virtualizes node locations. We did not study the security of VC routing. Overall, we believe these issues deserve separate studies due to their importance and related challenges.

References

[1] P. Bahl, V. Padmanabhan, RADAR: an in-building rf-based user location and tracking system, in: Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'00), vol. 2, Tel Aviv, Israel, March 2000, pp. 775–784.

[2] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, Wireless Networking 7 (6) (2001) 609–616.

[3] A. Boukerche, K. El-Khatib, L. Xu, L. Korba, An efficient secure distributed anonymous routing protocol for mobile and wireless ad hoc networks, Comput. Commun. 28 (10) (2005) 1193–1203.

[4] A. Boukerche, X. Li, An agent-based trust and reputation management scheme for wireless sensor networks, in: IEEE GLOBECOM, vol. 3, 2005.

[5] A. Boukerche, M.S.M. Notare, Behavior-based intrusion detection in mobile phone systems, J. Parallel Distrib. Comput. 62 (9) (2002) 1476–1490.

[6] N. Bulusu, J. Heidemann, D. Estrin, GPS-less low cost outdoor localization for very small devices, IEEE Personal Commun. Mag. 7 (5) (2000) 28–34.

[7] Q. Cao, T. Abdelzaher, A scalable logical coordinates framework for routing in wireless sensor networks, in: Proceedings of the 25th IEEE International Real-Time Systems Symposium (RTSS'04), Washington, DC, USA, 2004, IEEE Computer Society, Silver Spring, MD, pp. 349–358.

[8] S. Capkun, M. Hamdi, J. Hubaux, GPS-free positioning in mobile ad-hoc networks, in: Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS'01), Washington, DC, USA, 2001, IEEE Computer Society, Silver Spring, MD, p. 9008.

[9] A. Caruso, S. Chessa, S. De, A. Urpi, GPS free coordinate assignment and routing in wireless sensor networks, in: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05), vol. 1, March 2005, pp. 150–160.

[10] Q. Fang, J. Gao, L. Guibas, Locating and bypassing routing holes in sensor networks, in: Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04), vol. 4, March 2004, pp. 2458–2468.

[11] E. Felemban, C.-G. Lee, E. Ekici, R. Boder, S. Vural, Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks, in: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05), March 2005, pp. 2646–2657.

[12] R. Fonseca, S. Ratnasamy, J. Zhao, C.T. Ee, D. Culler, S. Shenker, I. Stoica, Beacon vector routing: scalable point-to-point routing in wireless sensornets, in: Proceedings of the Second Symposium on Network Systems Design & Implementation (NSDI'05), May 2005, pp. 329–342.

[13] S. Fotopoulou-Prigipa, A.B. McDonald, GCRP: geographic virtual circuit routing protocol for ad hoc networks, in: Proceedings of the First IEEE International Conference on Mobile ad hoc and Sensor Systems (MASS'04), 2004.

[14] D. Goldberg, Genetic Algorithm in Search, Optimization and Machine Learning, Addison-Wesley Publishing Company, Inc., Reading, MA, 1989.

[15] T. He, C. Huang, B.M. Blum, J.A. Stankovic, T. Abdelzaher, Range-free localization schemes for large scale sensor networks, in: Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking (MobiCom '03), New York, NY, USA, 2003, ACM Press, New York, pp. 81–95.

[16] T. He, J.A. Stankovic, C. Lu, T. Abdelzaher, SPEED: a stateless protocol for real-time communication in sensor networks, in: Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS '03), Washington, DC, USA, 2003, IEEE Computer Society, Silver Spring, MD, pp. 46–55.

[17] C. Karlof, Y. Li, J. Polastre, ARRIVE: algorithm for robust routing in volatile environments, Technical Report UCB//CSD-03-1233, University of California at Berkeley, Berkeley, CA, March 2003.

[18] C. Karlof, N. Sastry, D. Wagner, TinySec: a link layer security architecture for wireless sensor networks, in: Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys '04), New York, NY, USA, ACM Press, New York, 2004, pp. 162–175.

[19] C. Karlof, D. Wagner, Secure routing in wireless sensor networks: attacks and countermeasures, Ad hoc Networks J., Special Issue on Sensor Network Appl. Protocols 1 (2–3) (2003) 293–315.

[20] B. Karp, H.T. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in: Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom '00), New York, NY, USA, 2000, ACM Press, New York, pp. 243–254.

[21] Y.-J. Kim, R. Govindan, B. Karp, S. Shenker, Geographic routing made practical, in: Proceedings of the Second USENIX/ACM Symposium on Networked System Design and Implementation (NSDI'05), May 2005, pp. 217–230.

[22] L. Lazos, R. Poovendran, SeRLoc: secure range-independent localization for wireless sensor networks, in: Proceedings of the 2004 ACM Workshop on Wireless Security (WiSe '04), New York, NY, USA, 2004, ACM Press, New York, pp. 21–30.

[23] X. Lin, I. Stojmenovic, Location-based localized alternate, disjoint and multi-path routing algorithms for wireless networks, J. Parallel Distrib. Comput. 63 (1) (2003) 22–32 Special Issue on Routing in Wireless and Mobile Ad Hoc Networks.

[24] K. Liu, N. Abu-Ghazaleh, Virtual coordinate backtracking for void traversal in geographic routing, in: Proceedings of the Fifth International Conference on AD-HOC Networks & Wireless (AdHoc Now'06), August 2006.

[25] K. Liu, N. Abu-Ghazaleh, K.-D. Kang, JiTS: just-in-time scheduling for real-time sensor data dissemination, in: Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06), Washington, DC, USA, 2006, IEEE Computer Society, Silver Spring, MD, pp. 42–46.

[26] C. Lu, B.M. Blum, T.F. Abdelzaher, J.A. Stankovic, T. He, RAP: a real-time communication architecture for large-scale wireless sensor networks, in: Proceedings of the Eighth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'02), Washington, DC, USA, 2002, IEEE Computer Society, Silver Spring, MD, p. 55.

[27] D.J. Malan, M. Welsh, M. Smith, A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography, in: Proceedings of the First IEEE International Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON), Santa Clara, CA, October 2004.

[28] D. Malkhi, M. Reiter, Byzantine quorum systems, in: Proceedings of ACM Symposium on Theory of Computing, 1997, pp. 569–578.

[29] K.-D.K. Nael, B. Abu-Ghazaleh, K. Liu, Towards resilient routing in WSN, in: Proceedings of the First IEEE/ACM Workshop on QoS and Security in Wireless Networks (Q2SWinet 2005), 2005, pp. 71–78.

[30] R. Nagpal, H. Shrobe, J. Bachrach, Organizing a global coordinate system from local information on an ad hoc sensor network, in: Second International Workshop on Information Processing in Sensor Networks (IPSN'03), 2003.

[31] D.M. Nicol, M.E. Goldsby, M.M. Johnson, Simulation analysis of virtual geographic routing, in: Winter Simulation Conference, Washington, DC, USA, 2004, pp. 857–865.

[32] D. Niculescu, B. Nath, Ad hoc positioning system (APS), in: The IEEE Global Telecommunications Conference (GLOBECOM), vol. 5, 2001, pp. 2926–2931.

[33] D. Niculescu, B. Nath, Ad hoc positioning system (APS) using AOA, in: Proceedings of the 22nd Conference of the IEEE Communications Society (INFOCOM'03), 2003, pp. 1734–1743.

[34] C.E. Perkins, E.M. Royer, Ad hoc on-demand distance vector routing, in: The Second IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90–100.

[35] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, D.E. Culler, SPINS: security protocols for sensor networks, Wireless Networking 8 (5) (2002) 521–534.

[36] A. Rao, C. Papadimitriou, S. Shenker, I. Stoica, Geographic routing without location information, in: Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking (MobiCom '03), New York, NY, USA, 2003, ACM Press, New York, pp. 96–108.

[37] T. Roosta, M. Menzo, S. Sastry, Probabilistic geographic routing in ad hoc and sensor networks, in: International Workshop on Wireless Ad-hoc Networks, 2005.

[38] N. Sastry, U. Shankar, D. Wagner, Secure verification of location claims, in: Proceedings of the 2003 ACM Workshop on Wireless Security (WiSe '03), New York, NY, USA, 2003, ACM Press, New York, pp. 1–10.

[39] A. Savvides, C.-C. Han, M.B. Strivastava, Dynamic fine-grained localization in ad-hoc networks of sensors, in: Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking (MobiCom '01), New York, NY, USA, 2001, ACM Press, New York, pp. 166–179.

[40] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, D. Estrin, Data-centric storage in sensornets, SIGCOMM Comput. Commun. Rev. 33 (1) (2003) 137–142.

[41] USC ISI, Network Simulator 2 (2005).

[42] L. Venkatraman, D.P. Agrawal, Strategies for enhancing routing security in protocols for mobile ad hoc networks, J. Parallel Distributed Comput. 63 (2) (2003) 214–227.

[43] Y. Yu, R. Govindan, D. Estrin, Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks, Technical Report UCLA/CSD-TR-01-0023, Computer Science Department, UCLA, May 2001.

Ke Liu received his B.S. in Computer Science from Fudan University, Shanghai, China, in 2000, and his M.S from the State University of New York at Binghamton in 2005. He is currently working towards his Ph.D. in Computer Science. His research interests include wireless sensor networks: routing, scheduling and applications.

Nael Abu-Ghazaleh is an Associate Professor in the Department of Computer Science at the State University of New York at Binghamton. He received his Ph.D. and MS degrees in Computer Engineering from the University of Cincinnati. His research interests are in ad hoc networks, sensor networks, and distributed computing.

Kyoung-Don Kang is an Assistant Professor in the Department of Computer Science at the State University of New York at Binghamton. He received his Ph.D. from the University of Virginia in 2003. His research interest includes real-time data services including e-commerce and traffic/weather information service, wireless sensor networks, and wireless network security.