# Preserving Source Location Privacy in Monitoring-Based Wireless Sensor Networks

Yong Xi, Loren Schwiebert, and Weisong Shi
Wayne State University
Department of Computer Science
Detroit, MI 48202
{yongxi, loren, weisong}@wayne.edu

## Abstract

*While a wireless sensor network is deployed to monitor certain events and pinpoint their locations, the location information is intended only for legitimate users. However, an eavesdropper can monitor the traffic and deduce the approximate location of monitored objects in certain situations. We first describe a successful attack against the flooding-based phantom routing, proposed in the seminal work by Celal Ozturk, Yanyong Zhang, and Wade Trappe. Then, we propose GROW (Greedy Random Walk), a two-way random walk, i.e., from both source and sink, to reduce the chance an eavesdropper can collect the location information. We improve the delivery rate by using local broadcasting and greedy forwarding. Privacy protection is verified under a backtracking attack model. The message delivery time is a little longer than that of the broadcasting-based approach, but it is still acceptable if we consider the enhanced privacy preserving capability of this new approach. At the same time, the energy consumption is less than half the energy consumption of flooding-base phantom routing, which is preferred in a low duty cycle, environmental monitoring sensor network.*

## 1 Introduction

Wireless communication had gained more popularity in recent years. The application driven force behind the popularity is easy deployment and mobility. Besides the wide applications of wireless local network today, emerging applications of wireless communication include wireless sensor networks and Mesh Networks [4]. It can be easily seen that wireless networking will gain more popularity and vast information will be carried on wireless networks in the near future.

However, wireless communication media is a broadcast media, which poses a big challenge of how to protect information running on the network. Despite strong encryption of the data, wireless communication media still exposes some information about the traffic carried on the network. This is an inherent side effect of wireless communication. Mobility means that the communication is expected everywhere in the deployment area, which subsequently exposes the communication to possible attackers. Easy deployment means that there is certain openness in the protocol, which subsequently exposes some protocol information to possible attackers.

Location privacy is an important security issue. Loss of location privacy can enable subsequent exposure of identity information because location information enables binding between cyberspace information and physical world entities. For example, web surfing packets coming out of a home in a Mesh network enable an eavesdropper to analyze the surfing habits of one family if the source location of those packets can be determined.

In a wireless sensor network, location information often means the physical location of the event, which is crucial given some applications of wireless sensor networks. For example, in a battlefield, the location of a soldier should not be exposed if he initiates a broadcast query. In the *panda-hunter* problem, the location of the panda should not be exposed to hunters [8].

A wireless sensor network can be a low duty cycle network. Often, traffic has a strong correlation with a certain event at certain time. This gives big advantages to an eavesdropper since he does not need sophisticated techniques to discriminate traffic among different events. In this paper, we study the source location privacy problem under the assumption of one single source during a specific period. However, we need to point out that such a scenario can happen in a real wireless sensor network.

To preserve location privacy, we propose to use source and sink-based random walk for packet delivery. The sink first sets up a path through random walk which serves as a receptor. Each packet from a source is then randomly

1-4244-0054-6/06/$20.00 

forwarded until it reaches the receptor. At that point, the packet is forwarded to the sink through the pre-established path. A random walk greatly reduces the chance of packets being detected. Even if an eavesdropper happens to detect one packet, the next packet is unlikely to follow the same path, thus rendering the previous observation useless.

The reminder of the paper is organized into 5 sections. In Section 2, related work is presented. In Section 3, we show by an illustrated attack that randomness needs to be introduced carefully into the routing protocol. In Section 4, our implementation is described. In Section 5, simulation results are presented and discussed. In Section 6, we conclude our paper.

## 2 Related Work

Our work is inspired by [8, 6]. An application scenario of a wireless sensor network for monitoring a panda is presented. Enabling outside monitoring of a panda without exposing the location of the panda to hunters is proposed as the *Panda-Hunter problem*. Phantom routing is used for message delivery from the location of the panda to the sink for preserving its location privacy. The phantom routing algorithm is composed of two phases. In the first phase, the source initiates a random walk. In the second phase, the packet is being delivered through flooding or single path routing. In this paper, we specifically address a possible attack against the flooding-based delivery method.

The idea of using intersecting paths to deliver packets has been proposed in rumor routing [1]. In rumor routing, an event is known by some sensors in the small neighborhood of the event location. A query is sent through random walk. A usable delivery ratio is achieved by a large number of query random walks intersecting with each other. This is different from our approach. In our approach, both event and query source use random walk to advertise themselves. Also, our concern is to provide privacy protection; thus a more dynamic structure than rumor routing is needed.

In [10], asymptotics of three query strategies over a sensor network are discussed. Proofs are given that the probability of unsuccessful delivery using source and receiver driven 'sticky' Brownian motion decays much faster than using a single Brownian motion with increasing random walk length. ($t^{-5/8}$ vs $(log(t))^{-1}$ where $t$ is how long the Brownian motion has lasted) This result gives us a lower bound on the performance for our approach. In a real sensor network, the performance can be improved due to a limited size network. Also, in our approach, pure Brownian motion is not required for providing enough privacy protection.

In [3], the problem of hiding the location of the base station in sensor networks is discussed. An attack model of determining the base station location through traffic analysis is used. To hide the traffic pattern, randomly delaying the sending time is proposed to hide the parent-child relationship given a traffic rate model. Our work instead addresses the spatial pattern of the traffic.

In [5], the problem of sharing the location information without revealing the identity privacy in the mobile data collection applications, such as a cell phone periodically reporting its location, is discussed. Multi-target tracking algorithms can be used to identify each trajectory even when there is no identity information. A perturbation algorithm over multiple user paths is proposed to confuse the attacker. The algorithm takes advantage of the possible intersections of different paths and modifies location samples according to a nonlinear optimization solution. The artificially generated errors cause wrong trajectories being calculated by the attacker. This is different from our problem. In our model, the location information is not explicitly included in the packets.

## 3 What is Required for Preserving Source Location Privacy?

We consider an extreme case for preserving privacy in which there is traffic only from a single source in a network. This enables the eavesdropper to use just the spatial traffic pattern to compromise the source location privacy. This is a reasonable assumption. First, sensor networks are low duty cycle networks. The time spent for delivering a packet from the source to the sink can be much shorter than the source packet interval. Second, if the eavesdropper has access to the packet source information, he can isolate the source traffic from the rest of the traffic.

### 3.1 An Example Attack against the Flooding-based Phantom Routing

In this section, we illustrate a simulated attack against the flooding-based phantom routing. We assume that the eavesdropper has minimum physical capability, which is the ability to detect the presence of a radio transmission. Also, to get a good estimate of the source location, the eavesdropper consists of a group of devices distributed in the network. Each device at a different location is considered an observation point. However, as we argued before, the number of observations is limited. The purpose of the attack is to show that by using only a limited number of observation points the source location can be approximated without much effort.

At each observation point, the eavesdropper can record the time of a radio packet. The propagation speed can be modeled as a Gaussian distribution and is unknown. Also, the time when the algorithm begins to flood a packet is unknown. So, the parameters to be estimated comprise the following tuple: $(x, y, v, t)$, where $(x, y)$ are the coordinates of the location where flooding begins, $v$ is the propagation speed, and $t$ is the time when flooding begins. Suppose that

the coordinates of each observation point are $(x_i, y_i)$ and the packet is observed at time $t_i$. The true distance between an observation point and the flooding source is:

$$D_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} \quad (1)$$

The distance can also be written as:

$$VD_i = v(t_i - t) \quad (2)$$

Ideally, at each observation point we have $D_i = VD_i$. However, to estimate those four parameters, multiple observations at different locations are needed to solve the equation. Due to noise, the estimates at each observation will not be consistent. To find the optimal solution, we use the mean square error approach. We minimize the following formula:

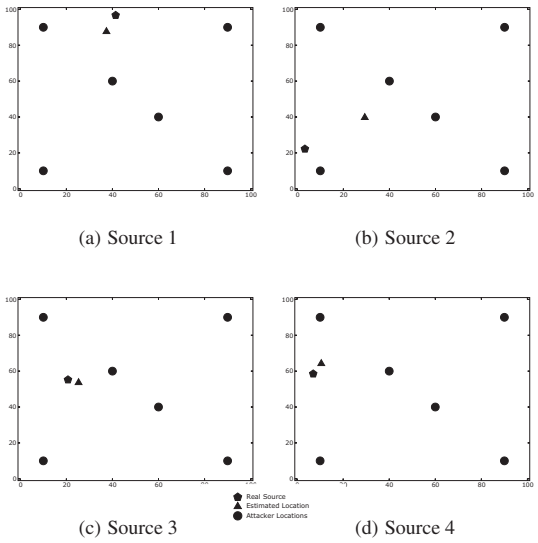
$$\sum |D_i - VD_i| \quad (3)$$

Ideally, four observation points should be enough for this purpose. However, in the simulation, we found that using six observation points yields much better estimates. Using six observation points compared with using four observation points is still acceptable. So, we present the simulation results with six observation points only.

To illustrate this attack, we have implemented the flooding-based phantom routing algorithm with TOSSIM [7]. We vary the number of hops during the random walk phase to check how this parameter affects the attack. The attack is being run over a network of 5000 nodes. We chose a large network size to show that even a large network can be susceptible to this attack. It's hard to preserve source location privacy in a small network under the assumption of only one single traffic existing in the network during a specific period.

We define the estimation error as the distance between the estimated location and true location. To measure the effectiveness of the attack, we fixed attackers at six locations in the network and varied the location of the source. The simulated network spans a rectangular area of size $100 \times 100$. The communication range of every sensor is 2.25. The six locations of attackers are $(10, 90)$, $(10, 10)$, $(90, 10)$, $(90, 90)$, $(40, 60)$, and $(60, 40)$. The choice of the six locations is rather arbitrary provided that they are relatively far from each other and have good coverage of the network. Note that the chosen locations are not necessarily close to the real source. Figure 1 shows the estimation errors for different scenarios within a period in which 50 source packets were sent out. Table 1 shows the estimation errors and the summations of mean square errors.

We deliberately return very large cost values for unreasonable solutions so that the optimization can converge faster. For example, scenario 2 in Table 1 has a large cost value. The reason is that the real location is outside the convex set of the observation points while the optimization is




(a) Source 1 (b) Source 2

(c) Source 3 (d) Source 4

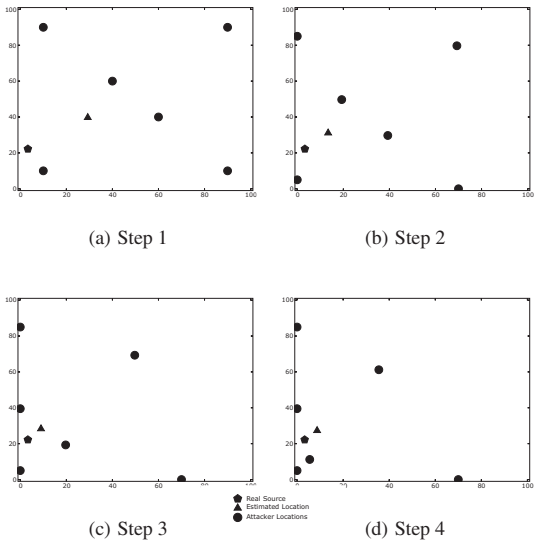
**Figure 1. Estimation Results for Four Sources**

**Table 1. Estimation errors and Mean-square errors**

| Scenario | Estimation Error | Mean-square Error |
|---|---|---|
| 1 | 10.0 | 2345.6 |
| 2 | 31.3 | $2.9 \times 10^{34}$ |
| 3 | 4.9 | 1588.2 |
| 4 | 6.7 | 2319.7 |

trying to find some point within this convex set. We adopt the following strategy to overcome this limitation. An inaccurate estimate has a very large cost value, which can be used by the eavesdropper to trigger the movement of the observation points. To illustrate this strategy, we moved the center of the original observation points toward the estimated location and re-estimate the location. However, during the moving process, if some observation points would move outside the network, we keep them at the boundary of the network. The whole process can be repeated. We use this strategy for the above example and the result is shown in Figure 2.

To investigate the effectiveness of the attack given different random walk steps, we vary the length of the random walk. During the simulation, we found that there are many local minimums in the topology we used above, where a node inside the network does not have any neighbor in one direction. This causes many packets to be dropped before reaching the flooding phase and deteriorates the estimate quickly. However, there is no suggestion on dealing with this problem in the phantom routing algorithm. To avoid the local minimum problem, we run the simulation on a network with 5000 sensors. The sensors are uniformly distributed in a $100 \times 100$ rectangle area. The increased density makes the




(a) Step 1 (b) Step 2

(c) Step 3 (d) Step 4

**Figure 2. Strategy to Close in on Source 2**

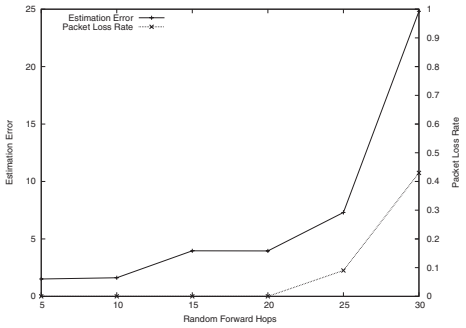



**Figure 3. Estimation error for different random walk hop counts**

local minimum a rare case.

Without loss of generality, we chose a source at $(25.0, 70.4)$. The random forward hop count is chosen for the values 5, 10, 15, 20, 25, and 30. The simulation results are shown in Figure 3. The estimation errors are larger for higher hop count values. However, even for a large hop count of 25, the estimate is still usable. Part of the reason that the estimation error gets worse is the way the phantom routing is designed. In our implementation, the forward directions are categorized according to sensors' $x$ coordinate. For random forward hop counts of 25 and 30, some of the packets are forwarded to the boundary of the network and dropped since there is no recovery mechanism defined. In Figure 3, this is shown as an increased packet loss rate. Since the source is located closer to one side of the network, only packets being forwarded to the closer side are lost. This causes the estimate to move toward the other side of the network. For those hop count values without packet loss, the increase in estimation error grows only linearly with the hop count and the growing speed is much slower than that of the hop count value. It shows that varying only the random forward hop count is not effective for providing better source location privacy.

### 3.2 Drawbacks of Flooding

Privacy is lost when the adversary is able to predict the source location within a reasonable period of time. In the above illustrated attack, the adversary can predict the approximate position of the source when a single packet is flooded. Although randomness is introduced through the random walk phase, the adversary can improve the prediction through statistical estimation.

Modeling the routing as a random process, the effectiveness of the adversary's strategy depends on how randomness is introduced and on how the adversary can sample this process. Given a known random process, every sample contributes to the adversary's estimation of the invariant parameters. In our case, the parameters are the $x$ and $y$ coordinates of the source. To deter the adversary from predicting the exact location of the source, we would like to slow down the speed at which the adversary can sample this process.

Assume that the source sends multiple packets to the sink over a period of time and uses consecutive sequence numbers to label those packets. The interval of packets received by the eavesdropper is defined as:

$$T = S_i - S_{i-1}, \tag{4}$$

where $S_i$ is the sequence number of the $i$th packet from the source arriving at the same physical location. $T$ is a random variable. The larger $T$'s mean, the longer it takes for the adversary to get a good enough estimate of the source location. Note that the sequence number is used only for analyzing. The packet does not have to have a sequence number.

Flooding is the worst method for protecting source location privacy in terms of $T$, which will take a fixed minimum value of 1 for all the locations in the network. Flooding enables the eavesdropper to accumulate information about the source location very quickly.

## 4 Greedy Random Walk

### 4.1 Random Walk and Source Privacy

The use of random walk is desired for protecting source location privacy. A random walk does not disclose any information about the source since the forwarding decision

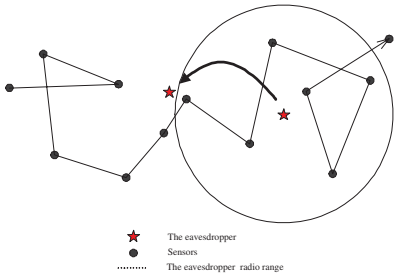
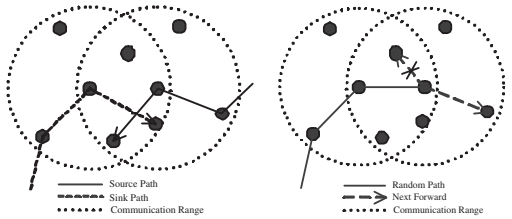



**Figure 4. The eavesdropper beats random walk within its radio range**




(a) Two Path Intersecting (b) Prevent Local Backtracking

**Figure 5. Non-planarity in Communication Graph**

is made locally and independent of the source location. In fact, an eavesdropper can not distinguish two random walks from two different sources. Using random walk also forces the adversary to use backtracking strategies, rendering the attack described in section 3 impossible.

However, a pure random walk tends to stay around the source [6]. Define the hitting time $T_a$ as the time when the Brownian motion path hits point $a$ for the first time. Here $a$ is any point other than the source. We have the following properties: [9]

$$P\{T_a < \infty\} = 1 \quad (5)$$

$$E[T_a] = \infty \quad (6)$$

If we put the source at 0 and the sink at $a$, it means that although the Brownian motion path will hit $a$ eventually, the average time it takes goes to infinite. It is not a desired result since it means average unbounded delivery time.

Since a Brownian motion path eventually hits $a$, it is important to see at what speed it converges to $a$. A recent work from Shakkottai [10] investigated this problem. The convergence is quantified as how fast the non-delivery probability decreases. It is shown that the probability decays as $(log(t))^{-1}$, where $t$ is how long the Brownian motion path has lasted. A more interesting result shows that if there is also a random walk from the sink at the same time, the probability of those two random walks not intersecting with each other decays as $t^{-5/8}$, which means that it is exponentially better than using only one random walk.

However, directly applying this approach is still not appropriate for practical applications because using random walk within the radio range of the eavesdropper is not useful to protect the source location privacy. For example, in Figure 4, the eavesdropper can move to the sensor from which it first hears the packet. Thus, the local random walk within the eavesdropper's radio range consumes extra energy and causes longer delivery time.

We propose a Greedy Random Walk (GROW) approach to address the above problems. In GROW, each time the sensor will pick up one of its neighbors which has not participated in the random walk. This way, the random walk is always trying to cover an unvisited area using a greedy strategy. Also, we eliminate local random walk and let both the source and sink initialize such a random walk to further improve the performance. The implementation of GROW is discussed in the next subsection.

## 4.2 GROW Algorithm

Previous analysis of random walk is based on a planar graph. However, this is not the actual communication graph in a wireless sensor network. If we treat the communication graph as a nonplanar graph during the implementation of the random walk, the probability of the source path and the sink path intersecting is much less than the previous asymptotic result. The scenario is shown in Figure 5(a). We use local broadcasting to solve this problem. Whenever a sensor forwards a packet, all its neighbors overhear this packet and create a route entry for the source pointing to the forwarding sensor. This does not require additional transmissions. Essentially the random walk is sticky not only for the sensors on the forwarding path but also for the neighboring sensors of this path. In effect, we build a pipe along the forwarding path.

The scenario not only exists between two paths, but also exists on a single random path itself. A random path might backtrack to itself after some time. However, we would like the path to extend as far as possible and as quickly as possible. In Figure 5(b), the sensor might forward the packet to one of its previous hop's neighbors. Such a forwarding decision is not good since the random walk does not make much progress. To prevent this case, we use a Bloom filter [2] to store all current neighbors in the forwarding packet. When the next hop randomly picks up one of its neighbors, it checks whether that neighbor is already in the filter. Given a limited number of neighbors, the probability of false positives can be made very small by using a reasonable size filter within

a packet. In other words, the packet will be forwarded to a sensor that has not seen the packet before with high probability.

However, the potential for backtracking still exists. The only possible way to prevent backtracking is to remember all the sensors which have already seen this packet. This is not realistic for a large scale network. Currently, we did not address this issue in this paper. Instead, we rely on increasing the random walk length to increase the coverage of the path. We are working on an improved method to address this issue. To decrease the chance of backtracking, each sensor keeps a Bloom filter to store those neighbors that have already participated in the forwarding. Each time a sensor is forwarding a packet, it will store the last hop from which the packet came and the next hop which it forwards the packet to. When the random walk backtracks to a sensor, it will choose one neighbor that has never forwarded the packet before. In this way, we hope to maximize the coverage given a fixed path length.

If the source and the sink are close to each other, the two random paths have a greater chance to intersect, thus the intersection points are closer to the source and the sink. This enables the eavesdropper to possibly trace the path. To prevent this from happening, we require a minimum path length of the source random walk.

Note that we do not assume any routing infrastructure in GROW for generality. If extra information is available, we can certainly use the information to improve the performance. For example, if geographical locations of sensors are known, it is easy to identify which part of the network has not been visited. Thus a more effective greedy forwarding based on this information can be used.

## 5 Performance Evaluation and Analysis

We implemented our algorithm in TOSSIM [7]. For comparison, the simulation is run over the same topology we used in Section 3. The topology is generated through uniformly deploying 5000 sensors within a rectangular area of $100 \times 100$. The communication range for each sensor is 2.25. We have tested our algorithm on several topologies; however, the algorithm does not perform significantly different on topologies generated with different random seeds. Thus, we present simulation results over only one topology.

### 5.1 Delivery Time

Ideally, the shortest delivery time is achieved if the packet is forwarded along the shortest path from the source to the sink. In flooding, the packet is forwarded approximately on this path since the propagation follows a wave pattern. This approximation can be proved through an induction process. However, due to space limitation, it is omitted here.

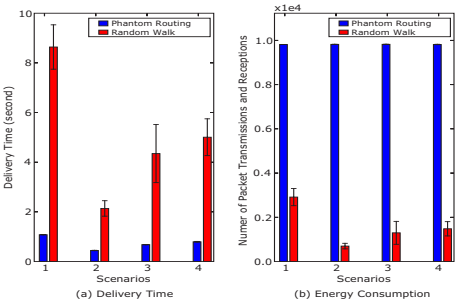



**Figure 6. Comparison between flooding-based phantom routing and Random Walk for Different Scenarios**

To provide source location privacy, it is necessary to relax the requirement for the delivery time. This is because if packets are always forwarded through the shortest path, it is easy for the eavesdropper to backtrack the path. There is certainly a trade-off between privacy and delivery time. We compare the delivery time between the flooding-based phantom routing and GROW. In Section 5.3, we show that privacy protection is provided by GROW.

For comparison, we use the same set of sources as those in Figure 1 and the same sink for both the flooding-based phantom routing and GROW. For the flooding-based phantom routing, we chose a rather conservative random walk hop count 5. For GROW, we fix the minimum path length of the source random walk to be 50. This value is chosen as on the scale of the network diameter.

Figure 6(a) shows the delivery time for both the flooding-based phantom routing and GROW. The delivery time is measured in seconds. Figure 7 is the cumulative distribution of delivery time for scenario 1. Although in scenario 1 the average delivery time is increased from 1 second to 9 seconds, over 50 percent of the packets are delivered within 5 seconds and 80 percent of the packets are delivered within 12 seconds. In other scenarios, the sources are closer to the sink. Thus the delivery time is considerably less.

### 5.2 Energy Consumption

In wireless sensor networks, packet transmission is generally the most power consuming operation. Packet reception also consumes significant energy, often on the same magnitude as packet transmission. To simplify the analysis, we assume that one packet reception consumes the same energy as one packet transmission and omit other energy consumption aspects of delivering a packet. The comparison is done on the total number of packet transmissions and receptions.

Figure 6(b) shows the comparison between the flooding-

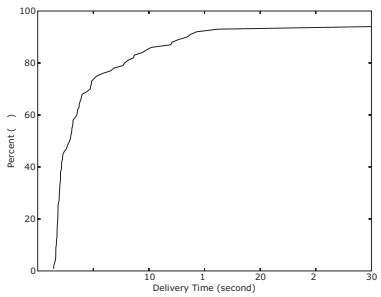



**Figure 7. Cumulative Distribution of Delivery Time in Scenario 1**

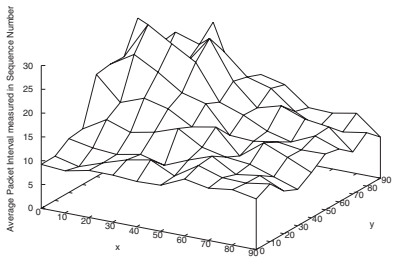



**Figure 8. Average Interval of Hearing a Packet at the Intersection Points for Different Sources across the network**

based phantom routing and GROW. The flooding-based phantom routing consumes the same energy regardless of the source. GROW consumes much less energy. Even in scenario 1, where the packet exists in the network much longer than that in the flooding-based phantom routing, the energy consumed by GROW is still less than half of the energy consumed by phantom routing. This shows the benefits of using random walks from both the source and sink. Although GROW tends to cover all the sensors, in practice it needs to cover only a small portion of the network to have the packet delivered.

### 5.3 Privacy Protection

In our approach, an eavesdropper needs to stay on the random path to track down the source. Since the random path from the sink is relatively stable, the best strategy for the eavesdropper is to start from the sink and backtrack the last hop each time he overhears a packet. However, to really get to the real source, the eavesdropper also has to backtrack the source random path. Since each packet from the source follows a different random path, the only relatively stable information which can be utilized by the eavesdropper is that the source path will intersect the sink path at some point. If the eavesdropper can not predict the next intersection point, he will miss the chance to make progress toward the source. To measure the privacy protection of our approach, we used the metric defined in Section 3 to calculate the mean interval between packets from the same source arriving at the same intersection point on the sink path.

To verify that we have consistent privacy for different sources across the network, we picked one hundred sources located approximately on a $10 \times 10$ grid over the network. The average intervals are shown in Figure 8. All average intervals are over 7 and the actual intervals are randomized. This makes it very hard for the eavesdropper to reliably catch a packet. Even if he stays at one location and eventually receives a packet, the last hop of the packet might be some location he has already visited since each time the packet follows a different random path. In such a case, he gets no new information and makes no progress toward the source.

To verify that our approach will not lead the eavesdropper to the source, we implemented a backtracking algorithm to simulate an eavesdropper. Originally, the eavesdropper stays near the sink. Once he detects a packet, he moves to the new location if he has never been there before. We run this algorithm over the simulated scenario 1. Figure 9 shows the trajectory of the eavesdropper during a period in which 250 source packets were sent out. The black dots are locations the eavesdropper has visited during this period. Although it looks like that the eavesdropper is approaching the source, the progress is slow. Ultimately the eavesdropper can reach the source since the greedy strategy eventually visits every sensor in the network. However, compared with the flooding-based phantom routing, in which the eavesdropper can compute very good estimate of the source location within a period of only 50 packets, the privacy protection is improved significantly.

## 6 Conclusions and Future Work

In this paper, we describe a possible attack against the flooding-based phantom routing. We propose GROW, a source and sink-based random walk as the alternative against this kind of attack. We improve the basic random walk by using local broadcasting and a Bloom filter. Simulation results show that it is practical to use our approach in a large scale wireless sensor network to protect source location privacy. Energy consumption is greatly reduced compared to the flooding-based phantom routing while there is only slight additional delay for message delivery. However, the delay is still acceptable. We believe that random walk is a basic approach for protecting source location privacy. However,

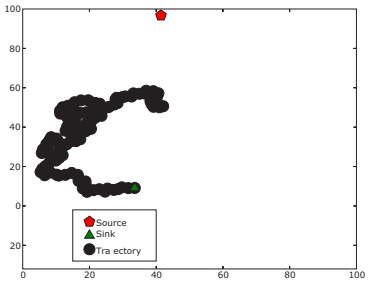



**Figure 9. Trajectory of the Eavesdropper over 250 packets**

there is still room for us to optimize the performance of this approach. Our future work is to find more efficient ways to build random paths.

## References

[1] D. Braginsky and D. Estrin. Rumor routing algorthim for sensor networks. In *the 1st ACM international workshop on Wireless sensor networks and applications*, 2002.

[2] A. Broder and M. Mitzenmacher. Network applications of bloom filters: A survey. In *Allerton Conference*, 2002.

[3] J. Deng, R. Han, and S. Mishra. Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks. In *IEEE International Conference on Dependable Systems and Networks (DSN 2004)*, 2004.

[4] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *ACM Mobicom*, 2004.

[5] B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *IEEE/CreateNet Intl. Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm)*, 2005.

[6] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk. Enhancing source-location privacy in sensor network routing. In *25th International Conference on Distributed Computing Systems (ICDCS 2005)*, 2005.

[7] P. Levis, N. Lee, M. Welsh, and D. Culler. Tossim: Accurate and scalable simulation of entire tinyos applications. In *the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, 2003.

[8] C. Ozturk, Y. Zhang, and W. Trappe. Source-location privacy in energy-constrained sensor network routing. In *the 2004 ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN) in conjunction with ACM Conference on Computer and Communications Security*, Oct. 2004.

[9] S. M. Ross. John Wiley & Sons, Inc, 2nd edition, 1996.

[10] S. Shakkottai. Asymptotics of query strategies over a sensor network. In *IEEE INFOCOM*, 2004.