Preserving Source and Destination Location Privacy with Controlled Routing Protocol

Rajorshi Biswas

Department of Computer and Information Sciences,
Temple University,
Philadelphia, PA, USA
E-mail: rajorshi@temple.edu

Jie Wu

Department of Computer and Information Sciences,
Temple University,
Philadelphia, PA, USA
E-mail: jiewu@temple.edu

Abstract: Efficiency in routing and security are two competitive design issues in wireless sensor networks. The most efficient and least secure routing protocol is shortest path routing. On the other hand, the most secure and least efficient routing protocol is random routing. In this paper, we propose the *controlled routing protocol*, a mixture of these two routing protocols that maintains a good balance between security and efficiency. Our proposed protocol is based on two principles: if all the messages do not follow the same path, then backtracking to the source node is not possible and when an adversary is very far away from the source and destination locations, then efficiency is more important than security. Based on these principles, we proposed the controlled routing protocol, in which the forwarding node forwards the message either to the node on the shortest path or a random neighbor with a variable probability. The probability of taking the shortest path increases by distance from the source and the destination node. In this paper, we also present our simulation results compared to other routing protocols.

Keywords: *Source Location Privacy, Security, Random Routing, Routing Protocol, Controlled Routing Protocol*

Reference

Biographical notes: Rajorshi Biswas is a PhD student of Computer and Information Sciences in Temple University, Philadelphia. He achieved his bachelor degree from Bangladesh University of Engineering and Technology, Bangladesh. He is currently doing research at Center for Networked Computing (CNC) which is focused on network technology and its applications. His research areas are Wireless Networks, Wireless Sensor Networks, Wireless Security, Cryptography, Cognitive Radio Networks etc.

Jie Wu is the Associate Vice Provost for International Affairs at Temple University. He also serves as Director of Center for Networked Computing and Laura H. Carnell professor. He served as Chair of Computer and Information Sciences from 2009 to 2016. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Service Computing and the Journal of Parallel and Distributed Computing. Dr. Wu was general co-chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.

1 Introduction

A wireless network consisting of spatially distributed sensor nodes that monitor physical or environmental conditions is called a wireless sensor network (WSN). A sensor node is an autonomous device that uses sensors to monitor physical or environmental conditions. This network system is being used for different applications like predictive maintenance, transportation, asset monitoring, health-care, safety, and security management. Wireless sensor networks are very useful in military organization. Military organization is the structuring of the armed forces of a state so as to offer military capability required by the national defense policy. Military organization is hierarchical. The building blocks of a typical military are commands, formations, and units. A command is a collection of units and formations under the control of a single commander. Formations and units deal with how commanders are selected and connected. Military operations in a battlefield without a communication infrastructure can be viewed as a wireless sensor network. For example, military units (e.g., soldiers, tanks, and drones), equipped with wireless communication devices, could form a wireless sensor network when they roam in a remote battlefield. Practical military operation systems need to deal with various security attackers. The attackers are assumed to have advanced equipment. This means that they have some technical advantages over the network nodes. Upon detecting an event, the attackers could determine the immediate sender by analyzing the strength and direction of the signal they received. Generally, the attackers will not be able to monitor the entire network; instead, they can efficiently monitor a specific area of the system. If the system remains constant for a certain amount of time, the attackers can gather various data about the system and launch attacks after further analyzing the gathered data.

So, the source and destination location privacy became an important issue in wireless sensor networks. Generally, messages follow the shortest path from the source to the destination. If a series of messages are transferred from the source to the destination, then an adversary can backtrack to the source by analyzing hop-by-hop message direction and strength. So, the main idea for maintaining source location privacy is to enforce messages to follow different paths. The idea is basically a moving target defense. A moving target defense (MTD) is a defense process where a defender continuously changes its attack surface. An attack surface is a set of resources which can be used by an attacker to attack. Changing the attack surface means changing the configuration of the resources that are used by the attacker. The changes in configuration make the system uncertain to the attacker. In this context, the attack surface includes the traffic pattern, routing protocol, signal direction, etc. Uncertainty adds more security to the defending system. The main goal of MTD is to make the defender uncertain to attacker by changing the configuration or parameters of a defender. In this scenario, MTD can be applied to routing protocols where the selection of the next node can be done randomly among neighbors. This solution provides the highest uncertainty but fails to guarantee the message delivery. We can refer to this as a random walk of messages. The traveled path of a message in a random walk is also very long. So, a mixture of shortest path routing and random walk can be a good choice. In the probabilistic random routing protocol, a message is forwarded using two strategies: random walk and shortest path routing. So, in each step, the forwarding node picks a random neighbor or the node on the shortest path to the destination. If the probability of random walk is high, then the message travels a long way. The probability can vary in messages depending on the security needed for the source/destination location.

In this paper, we present the controlled routing protocol, an improvement of the probabilistic random routing protocol. We added more control and improved efficiency by reducing the hop count of packets. The main idea of the controlled routing protocol is simple. When the forwarding node is far away from the source and the destination then taking the random walk strategy is unnecessary. So, we propose that probability of the shortest path routing will be increased by distance from the source or the destination. In other words, the probability of the random walk routing will be decreased by distance from the source or destination. So, the controlled routing protocol reduces the average hop count significantly compared to the probabilistic random routing protocol.

The rest of the paper is observed as follows. In Section 2, we will discuss some previous related works and their limitations. Section 3 defines the network and adversary models. We will discuss two types of adversaries in this section. In Section 2, we will present an existing work and our proposed controlled routing protocol. We will present our approximation of probability distribution of messages and expected path length in Section 5. In Section 6, we will discuss the security provided by our proposed algorithm in some quantitative ways. In Section 7, we will present our simulation results and show that our proposed protocol is better than the existing probabilistic random routing protocol.

2 Related Works

There are a lot of works on source location privacy/security. The source location privacy originated from Chaum's mixnet [1] and DC-net [2] protocols. After that, onion routing [3], tor [4], tarzan [5], morphmix [6], and buses for routing [7] are proposed for secure routing. These protocols provide unanimous message sending facilities in the network system. However, these approaches require public-key cryptography systems and are not suitable for WSNs. The adversary model in WSNs is different from the typical network system.

The adversary can move around and eavesdrop on the transmissions. So, encrypting the message cannot provide the source location privacy in WSNs. We will discuss some existing protocols that provide source location privacy in WSNs.

We find many existing works on source location privacy in WSN in [8]. In [9], the privacy of the source location is preserved by sending messages through remote nodes (called phantom nodes). Multiple phantom nodes [10] work as fake sources on behalf of the source node. The phantom nodes are randomly chosen and the diversity of the direction from the source node to the phantom node is maintained. Then, messages are delivered to the sink from the phantom nodes. In [11], variability in packet size is maintained using fake packets so that the original packet size is obfuscated. In [12], the routing consists of two phases. In the first phase, a certain hop random walk is done and the last node becomes the phantom node. In the second phase, flooding is initiated by the phantom node. As flooding offers the least security, adversaries backtrack to the phantom node and get stuck there. Routing through a random intermediate node (RRIN) is proposed in [13]. According to RRIN, intermediate nodes are far away from the source nodes that are randomly selected. Phantom node selection is quadrant-based according to the source and the destination, and maintain location diversity of selected relay nodes. In [14], authors proposed an anti-tracking source location privacy protocol called the path extension module (PEM). In the first round, nodes on the shortest path become fake sources in a probabilistic manner. In the second round, each fake source selects another neighbor and the neighbor becomes the fake source. Thus, the path is extended and branches are constructed. The adversary has a high chance of following fake traffic flow of branches. In [15], authors also proposed a fake source based routing protocol called SPENA for source location privacy. The source node activation technique protects the source location privacy from local and global adversaries. They also proposed failure node detection techniques in routing paths to divert adversaries to different paths. In addition, they used a node activation scheme to activate nodes at a particular time to confuse adversaries. Source location privacy is achieved by multiple redundant fog loops in [16]. The fog center node is selected after a certain hop random walk. Then, redundant messages are generated by the center and broadcasted to its neighbors. Multiple branch routing paths are generated within the fog, but only the routing path branch with the source node sends real data. Fogs are created to confuse the adversary and obfuscate the real traffic. Relay node selection is a key part of secure routing. In [17], a special region is defined and called the sink toroidal region (STaR). The source node randomly selects a relay node from the STaR area, in such a way that it is neither too close nor too far from the destination node. In the CASER[18] algorithm, the relay node is selected randomly among neighbors or the next node on the shortest path to the destination is selected. The probability of choosing the next node randomly can be different for each message. The challenge remains the balance between cost (in delay increase) and efficiency (in guard against the attacker). Secured routing can be also achieved through probabilistic flooding [19]. Every time a node receives a message it either broadcasts the message to its neighbors or discards the message. The probability of forwarding is predefined. However, this approach does not guarantee delivery. The authors in [20] proposed a sink location protection scheme based on the message sending rate adjustment (SRA). SRA achieved sink location privacy by controlling the packet sending rate of each node dynamically and balancing traffic over the entire network. This mechanism obfuscates the real traffic patterns, and thus the location of the sink is protected.

All the related works discussed above are based on some basic routing strategies like random walk, flooding, phantom routing, and shortest path routing. Some of them used fake message transmissions from fake sources to obfuscate the original traffic of the network system. Most of the previous works focus on the privacy of the source location only. Flooding, random walk, and fake message transmission techniques consume more energy. Phantom routing techniques are not secure enough because adversaries can analyze the direction of the source if the selection of the phantom node is not good enough. The path from the source to the phantom node is also vulnerable. Since random walk provides the most uncertainty to the routing path, it cannot guarantee the delivery of messages and the path length is also very high. So, there remains the challenge of providing a good balance between efficiency and security.

3 Network Model

In this section we will discuss the network model and the adversary model. In general, there is more than one sender and receiver in a WSN. For simplicity, we will assume one source and one destination throughout this paper.

3.1 The System Model

The system model is similar to the famous panda hunting problem. In the panda monitoring sensor network system, a lot of sensor nodes are homogeneously distributed throughout a large area. Whenever a panda is observed, the nearest node initiates sending a message to a destination node which keeps a record of the monitoring data. The sensor node keeps sending messages while observing the panda. On the other hand, the hunter is equipped with advanced technologies and can monitor the traffic of sensor nodes. He can also detect the location of the immediate sender with his equipment. Once the hunter eavesdrops on any message sending event, he moves to the immediate sender and

waits for sometime to eavesdrop on another event. If the series of messages follow same path, then the hunter is able to trace back to the source and hunt the panda. Our objective is to make the path uncertain so that it will be difficult enough for the hunter to trace back.

We make the following assumptions about our system:

1. **Uniform Node Distribution:** The network is composed of many uniformly distributed nodes throughout the area. The network is connected.
2. **Knowledge About Location:** Every node in the network knows the location of the source and the destination. So, every node can calculate the geographic distance to the source and the destination.
3. **Single Source and Single Destination:** For simplicity and ease of explanation, we assume the number of source nodes is one. If multiple nodes simultaneously identify the monitoring object (panda), the nearest to it acts as a sender and the others remain inactive. All messages go to a destination. Every node knows the location of the destination.
4. **Encrypted Message:** For security of the contents, messages are encrypted. Every node shares the encryption key and reads the contents of the message. The source and destination IDs are attached to each message. While decrypting a message, a node knows the source and destination IDs or locations of the messages.

3.2 The Adversary Model

The adversary is intelligent and equipped with the signal analyzing hardware. It is not harmful to the network system. It is only interested in identifying the location of the monitoring object. We assume the following characteristics of the adversaries.

1. The adversary is equipped with unlimited power storage and memory. So, it can travel anywhere at anytime without worrying about energy. The memory storage is used for storing important travel information and observations.
2. The adversary is passive in nature. It does not interfere with the proper functioning of the network, which means it does not modify messages, change routes, or destroy sensor nodes.
3. The adversary is equipped with a directional antenna so that it can analyze the direction of the signal and locate the immediate transmitter.

Initially, the adversary can remain anywhere in the area. Whenever the adversary eavesdrops on any transmission, it analyzes the signal strength and direction and locates the immediate sender. Then, it travels to that location and waits to eavesdrop on another signal. The process continues until it finds the source, destination, or monitoring object.

Based on capability, we can classify the adversary model into two classes: local adversary and global adversary. The capability of monitoring the network area is limited for local adversary. It can view the network traffic and other events of a limited area near its location. On the other hand, global adversaries can view the network traffic and events of the whole area. In this paper, we will discuss the security of the source and destination locations for local adversary model.

3.3 Security Analysis of Shortest Path Routing and Flooding

In shortest path routing, the path from the source to the destination is deterministic. Every message follows the same path. If every message follows the same path, then there is a high chance that the adversary will backtrack to the source. The scenario is described in Figure 1. In Figure 1, source node 5 sends M_1, M_2, M_3, and M_4 messages sequentially to destination node 20. All messages follow the shortest path. At time T_1, node 14 sends the message M_1 to destination node 20. The adversary, staying near node 14, eavesdrops on the message M_1. With advanced equipment, the adversary calculated the location of the immediate sender 14 by analyzing the strength and direction of the signal. While the transmission of M_1 was going on, the adversary moves to node 14. After that, the adversary eavesdrops on the message M_2 at time T_2 and moves to node 10. Then, after eavesdropping on M_3 at time T_3 it moves to node 11. Finally, after eavesdropping on M_4 at time T_4, it identifies the source location.

Flooding is being used in many sensor networks for the dissemination of information. In flooding, the source initiates a message and broadcasts it to its neighbors. If a neighbor did not broadcast the message earlier, will broadcast the message to all of its neighbors. This kind of routing has the worst security for the source location. The scheme is actually sending messages through the shortest path to all other nodes in the network. So, the adversary can backtrack to the source from any location in the area.

To prevent the backtracking of the adversary, the series of messages shouldn't follow the same path. If the message M_2 didn't follow the same path as M_1, then the adversary wouldn't have been able to eavesdrop on the message M_2 at time T_2. Consequently, the adversary would be stuck at node 14.

4 Routing Protocols for Source and Destination Location Privacy

In this section, we will present the probabilistic random routing protocol and our proposed routing protocol to obtain the source and destination location privacy.

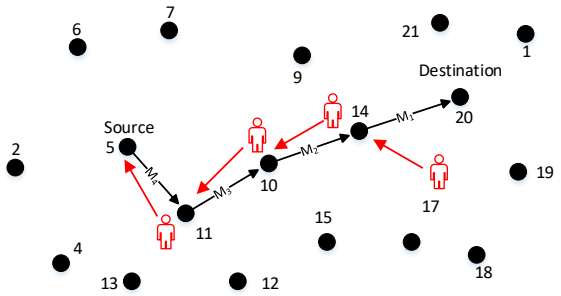


Figure 1 Shortest path routing and backtracking of adversary.

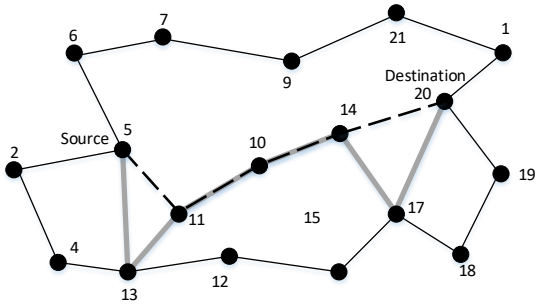


Figure 2 Controlled routing protocol.

4.1 Probabilistic Random Routing Protocol

The probabilistic random routing protocol (PRRP) is proposed as cost-aware secure routing protocol (CASER) in [18]. In this routing protocol, the forwarding node selects the next node based on any of the two routing strategies: shortest path routing, and random routing. The forwarding node selects the next node as either the node on the shortest path or a random neighbor. The probability of choosing the strategy remains fixed throughout the path. The probability of choosing any strategy can vary for different packets based on their characteristics and security needed. Assume the probability of selecting random routing is α. So, if a packet travels N steps to reach the destination then ideally αN steps are taken randomly and $(1-\alpha)N$ steps are taken from the shortest path. The algorithm is shown in Algorithm 1.

Algorithm 1 Probabilistic random routing protocol (PRRP)

Input: A node n, and a control variable α.

1: Compute A where $A[u][v]$ denotes the next node on shortest path from u to v.
2: $Q \leftarrow$ message queue of n.
3: **while** $Q \neq \emptyset$ **do**
4: Dequeue message m from Q.
5: Select a random number $r \in [0, 1]$.
6: $m_d \leftarrow$ destination of packet m.
7: **if** $r < \alpha$ **then**
8: Select next node randomly from neighbors of n and forward message m.
9: **else**
10: Select next node $(A[n][m_d])$ from shortest path routing table and forward message m.

4.2 Proposed Routing Algorithm

To achieve a better balance of cost without compromising the security compared to PRRP, we propose the controlled routing protocol (CRP) which also combines random routing and shortest path routing. Any node forwarding a packet will choose the next node either randomly from its neighbors or the node remains on the shortest path to the destination. Let's denote the probability of choosing the next node randomly by α. So, the probability of selecting the node that remains on the shortest path is $1-\alpha$. In PRRP α remains constant over all the paths from the source to the destination, but in CRP α varies based on distance from the forwarding node to the source or the destination. The algorithm is shown in Algorithm 2.

$$\alpha = f(d), d = \min\{d_s, d_d\}. \tag{1}$$

Here, d_s is the distance of the forwarding node from the source and d_d is the distance of the forwarding node from the destination. Let's call the $f(d)$ a randomness control function. $f(d)$ can be any function which decreases while d increases. $f(d)$ can be defined as follows:

$$f(d) = \omega + (1-\omega)\gamma\sqrt{\frac{6d}{2d^2+3d+1}} \tag{2}$$

Where ω and γ are predefined constants. ω denotes the preserved randomness level and γ denotes the effectiveness of the distance. We refer to both ω and γ as the control variables of the controlled routing protocol. The last part $(\sqrt{\frac{6d}{2d^2+3d+1}})$ of $f(d)$ is distance sensitive and decreases while d increases. The last part is basically a reciprocal of root mean square of distances of each node on the shortest path from the source (or destination whichever is closer to current node) to the forwarding node, multiplied by square root of forwarding node distance. So, the distance sensitive part (DSP) is as following:

$$DSP = \sqrt{\frac{d}{\sum_{i=1}^{d} i^2}} \times \sqrt{d} = \sqrt{\frac{6d}{2d^2+3d+1}} \tag{3}$$

We choose the above DSP because the value of DSP decreases smoothly w.r.t. distance. The smoothness of DSP is important as the diversity of message around the source or the destination depends on it. The more the DSP is smooth the more the diversity of messages.

The controlled routing algorithm is slightly different than the probabilistic routing algorithm like CASER[18]. Instead of a fixed probability throughout the path from the source to the destination, it uses a variable probability for random routing. The probability of random routing is higher if the source or destination is

Algorithm 2 Controlled routing protocol (CRP)

Input: A node n, and control variables ω and γ.

Compute D where $D[u][v]$ denotes the shortest path distance from u to v.
Compute A where $A[u][v]$ denotes the next node on shortest path from u to v.
$Q \leftarrow$ message queue of n.
while $Q \neq \emptyset$ **do**
  Dequeue message m from Q.
  $s \leftarrow$ source of message m.
  $d \leftarrow$ destination of message m.
  $d \leftarrow min\{D[s][n], D[n][d]\}$.
  $\alpha \leftarrow \omega + (1-\omega)\gamma\sqrt{\frac{6d}{2d^2+3d+1}}$.
  Select a random number $r \in [0, 1]$.
  if $r < \alpha$ **then**
    Select next node randomly from neighbors and forward message m.
  else
    Select next node ($A[n][m_d]$) from shortest path routing table and forward message m.

closer to the forwarding node. So, nodes close to the source or the destination forward more randomly than nodes that are far away. Nodes which are far away from the source or the destination mostly route according to the shortest path routing. The algorithm is shown in Algorithm 2.

In Figure 2, node 5 is the source and node 20 is the destination. The shortest path from 5 to 20 is (5, 11, 10, 14, 20) which is denoted by dotted lines. According to the controlled routing protocol, the message may travel nodes 5, 13, 11, 10, 14, 17, and 20. The path is denoted by thick gray lines. Source 5 sends the packet to its random neighbor 13. From node 13 to 14, the message takes the shortest path. As 14 is close to destination 20, it forwards the packet to node 17 which is chosen randomly from node 14's neighbors and node 17 forwards to destination node 20. In this routing protocol, random routing increases when the forwarding node is close to the source or the destination. Random routing near the source and the destination secures the location of them and the shortest path routing farthest from them increases efficiency.

5 Probability Distribution

According to our proposed algorithm, an infinite number of paths are possible from a source to any node. For simplicity, we assume all the nodes are in a grid and only left, right, up and down forwarding are possible. To find the probability distribution, we also assume that the source S located on $(0, 0)$ and the destination node $D(0, L)$ are very far away from the source location. We want to calculate the probability that a packet is located at any node $A(x, y)$ after N steps. For simplicity, we assume α changes according to the distance from the source only. The behavior of the algorithm near the destination is identical to near the source. So, the probability distribution near the destination will be same as near the source.

The routing of messages can be divided into two separate stochastic processes: one process is related to going left or right, and another process is related to going up or down. For the left-right stochastic process, we assume the probability of going right is p and probability of going left is q. p and q are fixed throughout the path. After N_x steps the probability of the packet staying at x steps away from the source S is given by:

$$P_x(x, N_x) = \frac{N_x!}{(\frac{N_x+x}{2})!\,(\frac{N_x-x}{2})!} p^{\frac{N_x+x}{2}} q^{\frac{N_x-x}{2}}, \tag{4}$$

where, $N_x + x$ must be an even number and $N_x \geq x$.

For the up-down stochastic process, we assume the probability of going up is u and down is d which are fixed throughout the path. After N_y steps the probability of the message staying at y steps away form the source S is given by:

$$P_y(y, N_y) = \frac{N_y!}{(\frac{N_y+y}{2})!\,(\frac{N_y-y}{2})!} u^{\frac{N_y+y}{2}} d^{\frac{N_y-y}{2}}, \tag{5}$$

where $N_y + y$ must be an even number and $N_y \geq y$.

Combining these two equations, we can approximate the probability that a packet is located at any node $A(x, y)$ after N steps.

$$P(x, y, N) \approx \sum_{N_x=x}^{N-y} \frac{N_x!}{(\frac{N_x+x}{2})!\,(\frac{N_x-x}{2})!} p^{\frac{N_x+x}{2}} q^{\frac{N_x-x}{2}} \times \sum_{N_y=y}^{N-x} \frac{N_y!}{(\frac{N_y+y}{2})!\,(\frac{N_y-y}{2})!} u^{\frac{N_y+y}{2}} d^{\frac{N_y-y}{2}}$$

According to our proposed routing algorithm, the value of p, q, u, and d are not constant all over the path. We approximate the following values of p, q, u, and d for different locations. Let's assume the probability of random routing α is divided into two parts for two stochastic processes. The probability of random routing is α_x for the left-right stochastic process and α_y for the up-down stochastic process.

$$\begin{aligned} \alpha_x &= \omega + (1-\omega)\gamma\sqrt{\frac{6x}{2x^2+3x+1}}, \\ \alpha_y &= \omega + (1-\omega)\gamma\sqrt{\frac{6y}{2y^2+3y+1}} \end{aligned} \tag{6}$$

p, q, u and d can be determined from Equation 6. For the first coordinate $(x > 0, y > 0)$:

$$\begin{aligned} p = (1-\alpha_x) + \frac{1}{2}\alpha_x &= 1 - \frac{1}{2}\alpha_x, \\ q = 1 - p &= \frac{1}{2}\alpha_x, \\ u = \frac{1}{2}\alpha_y, \;\; d = 1 - u &= \frac{1}{2}\alpha_y \end{aligned} \tag{7}$$

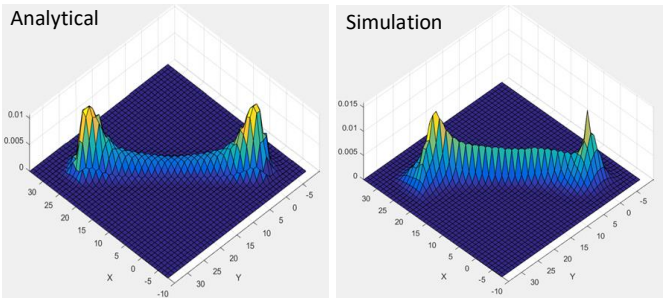


Figure 3 Analytic distribution ($P(x, y, N)$) and distribution from simulation.

The distribution of $P(x, y, N)$ is a binomial distribution. Figure 5 shows the comparison between our approximated distribution and the distribution from the simulation. We choose $\omega = 0$ and $\gamma = 1$ as this setting shows the maximum impact of distance in choosing routing strategy. The source location is (0, 0) and destination location is (25, 25). Our approximated distribution shows a 73% similarity with the simulation result.

5.1 Expected Path Length

To approximate the expected path length, we also assume two different stochastic processes for left-right and up-down forwarding. If a message travels N_x steps along the X axis, then the expected distance along the X axis is x:

$$N_x = \frac{x}{1-\alpha_x} = \frac{x = N_x(2p-1) = N_x(1-\alpha_x),}{1-(\omega+(1-\omega)\gamma\sqrt{\frac{6\bar{x}}{2\bar{x}^2+3\bar{x}+1}})} \quad (8)$$

Similarly, for the up-down stochastic process, if the packet travels N_y steps along the Y axis, then the expected distance along the Y axis is y:

$$N_y = \frac{y}{1-(\omega+(1-\omega)\gamma\sqrt{\frac{6\bar{y}}{2\bar{y}^2+3\bar{y}+1}})} \quad (9)$$

Where $\bar{x}$ and $\bar{y}$ are the approximated effective distances of location (x, y) from the source along X and Y axes, respectively. We approximated the following expression from our simulation result.

$$\bar{x} = \frac{x}{ln(x)},\ \bar{y} = \frac{y}{ln(y)} \quad (10)$$

The traffic pattern near the source and the destination are similar because the probability of random routing depends on the closest distance from the source or the destination. Therefore, the way α changes between the source to the middle point is similar to the the way α changes between the destination to the middle point. The total steps it takes to travel to the location (x, y) is $N(x, y)$.

$$\begin{aligned} N(x,\,y) &\approx 2N_{\frac{x}{2}} + 2N_{\frac{y}{2}} \\ &\approx \frac{x}{1-(\omega+(1-\omega)\gamma\sqrt{\frac{\overline{x/2}}{2(\overline{x/2})^2+3(\overline{x/2})+1}})} \\ &+ \frac{y}{1-(\omega+(1-\omega)\gamma\sqrt{\frac{6\overline{y/2}}{2(\overline{y/2})^2+3(\overline{y/2})+1}})} \end{aligned} \quad (11)$$

We conducted a simulation with $\omega = [0, 0.9]$ and $\gamma = [0, 1]$ and compared the simulated average path length with the approximated path length. We found that our approximated expected path length is 96% accurate. Some of the results are shown in Table 1.

6 Security Analysis

In this section, we will analyze the security provided by CRP. In [21], authors introduced criteria to quantitatively measure the source location privacy for WSNs. We will discuss the security of CRP in terms of SDI and probability of backtracking to the source or the destination.

6.1 Source-location Disclosure Index (SDI)[21]

SDI measures, from an information entropy point of view, the amount of source-location information that one message can leak to the adversaries.

Firstly, according to our assumption, the messages are encrypted and only the nodes can decrypt and see the message content. So, the ID of each node is not visible to the adversary and cannot link one message to another. So, the source/destination identities are preserved.

Secondly, the probability of random routing is determined by α which depends on the minimum distance from the source or the destination to the current node and some predefined control variables (ω and γ) . As the ω increases the random routing throughout the path also increases and the path becomes more unpredictable. For $\omega > 0$ or $\gamma > 0$ at each intermediate node, a message source or destination direction cannot be determined by analyzing the direction of any transmission. A node which is very far away from the source and the destination is more likely to route by the shortest path algorithm. As the shortest path is not the straight line connecting the source and the destination the direction cannot be determined. At that location, the adversary will able to backtrack some parts of the paths as some of the messages will travel the same parts of the path. When the adversary gets closer to the source or the destination random routing increases and the path becomes more uncertain.

Finally, the adversary will not able to analyze the traveled hop count as the path is not deterministic. In that sense, we can conclude that CRP has $SDI \approx 0$.

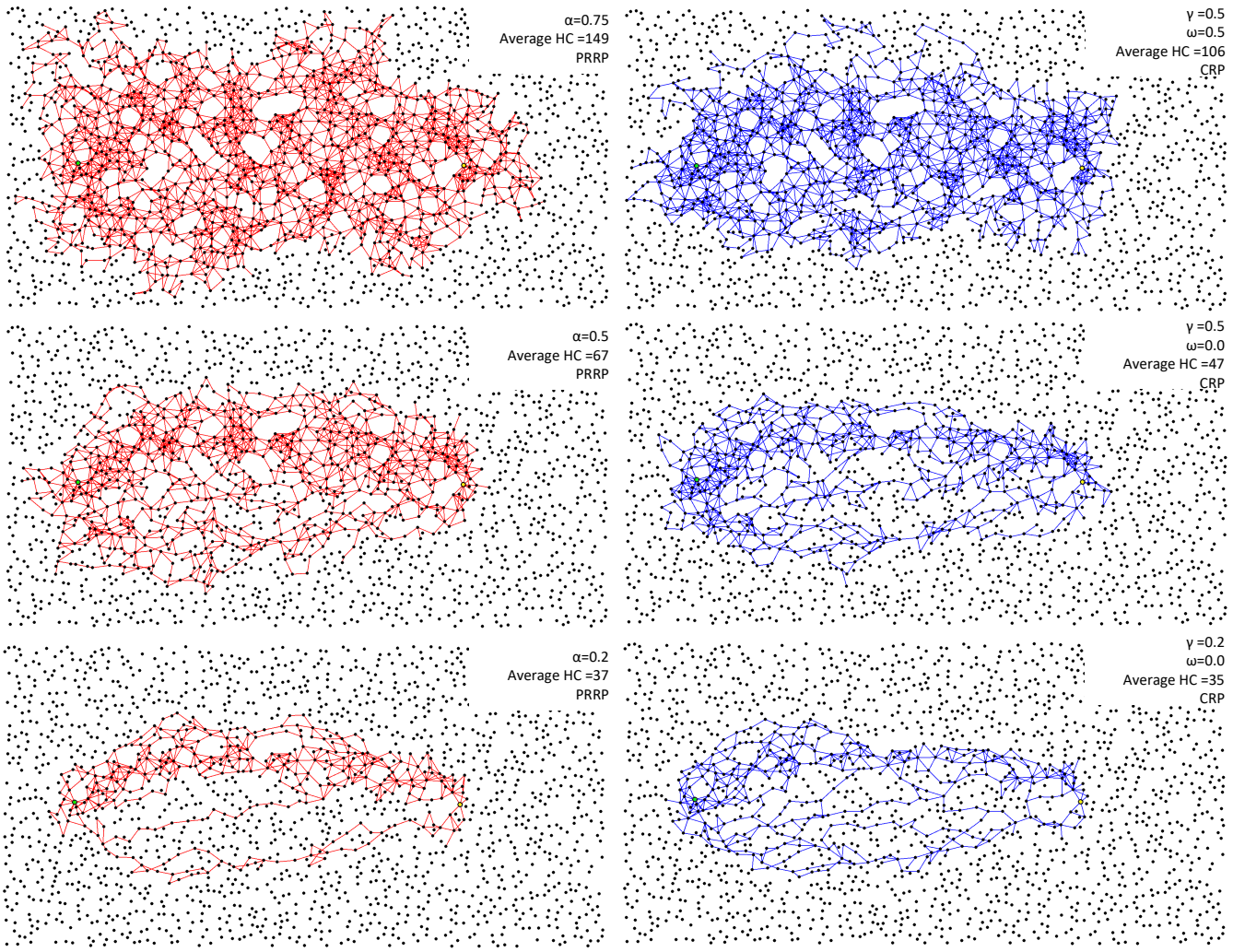


Figure 4 Simulation screen-shot of PPRP and CRP.

Table 1 Path length from approximation and simulation.

ω	γ	Approximation	Simulation	Accuracy %
0	0.0	40	40	100
0	0.4	55.85	54.32	97.18
0	0.8	92.52	87.68	94.48
0	1.0	137.74	141.72	97.19
0.4	0.0	66.67	70.66	94.35
0.4	0.4	93.09	97.02	95.95
0.4	0.8	154.2	144.02	92.93
0.4	1.0	229.56	230.42	99.63
0.8	0.4	279.27	312.86	89.26
0.8	0.8	462.61	460.34	99.51
0.8	1.0	688.68	660.12	95.67

6.2 Probability of Backtracking to Source or Destination

In CRP, the probability of shortest path routing is highest at the middle point on the shortest path from the source to the destination. In that location, the probability of message transmission is very high. So, it is likely that the adversary will manage to find that location easily. We assume the adversary is staying at that point. Now, we want to calculate the probability of finding the source or destination from that particular location. Let's assume the average number of neighbors of each node is K and the path from source S to destination D is $(N_1, N_2, ...N_n)$. The adversary remains near node $N_{n/2}$. So, the probability of a packet transmitted from node $N_{n/2-1}$ to $N_{n/2}$ is the probability of backtracking to node $N_{n/2-1}$ from $N_{n/2}$. The probability of backtracking to node $N_{n/2-1}$ from $N_{n/2}$ is:

$$P(N_{\frac{n}{2}-1}) = (1 - \alpha_{\frac{n}{2}-1}) + \frac{\alpha_{\frac{n}{2}-1}}{K} \tag{12}$$

Thus, the probability of backtracking to the source can be obtained from following:

$$P(S) = \prod_{i=1}^{\frac{n}{2}} ((1 - \alpha_i) + \frac{\alpha_i}{K}) \tag{13}$$

The probability of backtracking to destination will be same as probability of backtracking to the source. If the average neighborhood $K = 4$, $\omega = 0$, $\gamma = 1$, and

the source and destination are 20 hops away from each other, then the probability of backtracking to the source or destination is 0.037 which is very small. So, CRP provides enough security to the source and the destination.

7 Performance Evaluation and Simulation Results

In this section we will present our experimental settings and simulation results.

7.1 Experimental Settings

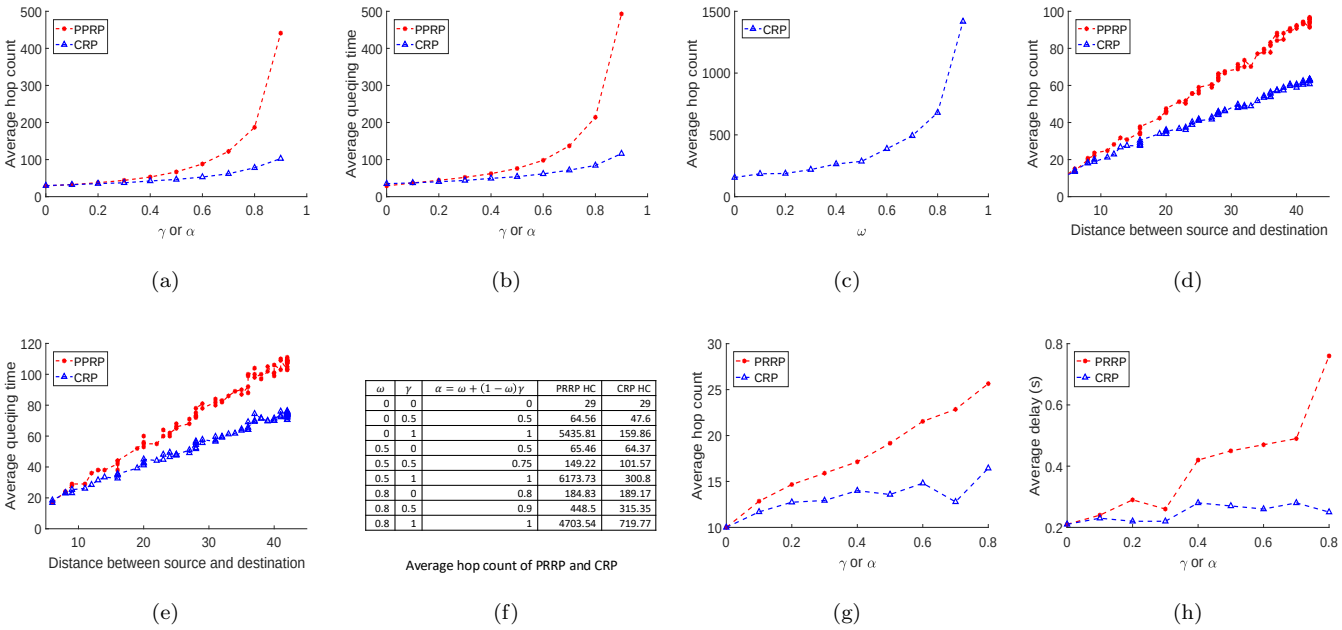
We conducted our simulation using JAVA programming to compute the average hop count and the average queuing time of a message. We define queuing time as the time a packet remains in the queue of a node. The queuing time is proportional to the position of the message in a queue. We calculated the positions in the queue of a message in all nodes. We took the average of the total queuing time of all the messages. We took a 1920×1080 area to conduct our simulation. We divided the whole area into 70×70 blocks to distribute nodes. We distributed a maximum of 8 nodes on average to each block. As every block has an almost equal number of nodes, nodes are located uniformly throughout the area. We assumed circular coverage of node with a fixed radius. So, a couple of nodes are neighbors if their geographical distance is less than the radius. We choose the radius of coverage to be 30. Our system has 1,758 nodes and 7,767 edges. We kept a message queue in each node. When a message is forwarded from one node to another, it just dequeues from its message queue and enqueues in anothers' message queue. The position of a message in a message queue determines the delay of message forwarding. We assumed the fixed packet size and the total time needed to deliver a packet is its queuing time multiplied by the time needed to transmit a packet. We will evaluate the performance of the controlled routing protocol in terms of average hop count and average queuing time.

7.2 Performance Evaluation

Figure 4 depicts the comparison of the PRRP and the CRP. We conduct our simulation with the aforementioned settings and with 1000 packets routing. We calculated the average hop count and average queuing time when all the packets are delivered to the destination. We considered different values ω, γ (for CRP) and $\alpha = \omega + (1 - \omega)\gamma$ (for PRRP) and calculated the average hop count and queuing time using both the PRRP and the CRP. Each time, CRP performed better than PRRP. So, in terms of hop count and queuing time, the CRP works much better than the PRRP.

We can also observe the difference between PRRP and CRP in traffic patterns. The followed paths of the messages in PRRP are more diverse than CRP. This is because PRRP uses a constant probability for random routing. So, the paths continue to be diverse at a constant rate. On the other hand, the probability of random routing decreases by distance from the source or the destination. So, the diversity decreases while the packet moves away from the source. For this reason, we observe less diversity in the packets in CRP than PRRP. Figure 5(a) shows the average hop count of PRRP and CRP w.r.t. control variables γ (for CRP) and α (for PRRP). We kept $\omega = 0$ to observe the difference between PRRP and CRP. We can observe that the average hop count remains the same at $\gamma = 0$ as messages in both PRRP and CRP follows the shortest path routing. When γ increases, random routing increases and PRRP has more hop counts than CRP. The hop count in CRP increases almost linearly. On the other hand, the hop count increases exponentially in PRRP w.r.t. α. Figure 5(b) shows the comparison of the average queuing times between PPRP and CRP. The average queuing time shows behavior similar to the hop count. We also conducted simulations with varying $\omega = [0, 0.9]$, keeping $\gamma = 1$. The impact of ω in CRP is similar to the impact of α in PRRP. The average hop count increases exponentially by ω in CRP. When the value of ω is close to 1, PRRP and CRP show almost similar behaviors. We conducted simulations by varying the control variables $\omega = [0, 0.9]$ and $\gamma = [0, 0.9]$. We found a significant amount of reduction in the hop count. Some of the results are given in Figure 5(f). We did another simulation of PRRP and CRRP with $\omega = 0$, $\gamma = 0.5$, $\alpha = 0.5$, and a varying shortest path distance between the source and the destination to see the difference. We started with the shortest path distance from 5. At that point, PRRP performed better than CRP. This is because in CRP, the source picks a random neighbor to forward. So, whenever the distance between the source and the destination is very short, PRRP seems to perform better. In reality, the source and the destination are not close to each other, so we can ignore that case. When the distance is longer, we can observe from Figure 5(d) that CRP has a much smaller average hop count than PRRP. The difference in hop count increases by distance between the source and the destination. We can observe a similar kind of behavior for the average queuing time, shown in Figure 5(e). We found a 40% reduction in hop count and a 36% reduction in average queuing time in CRP compared to PRRP.

NS3 Simulation: We do simulation with 100 nodes in NS3 to see the behavior of delay of PRRP and CRP. Nodes are distributed in grid in a square area. We set the transmission power according to the distance between nodes so that only left, right, up, and down nodes are reachable. Nodes communicates each other by 802.11b protocol. The source node continuously creates UDP packets. We set the source and destination to be on the (3,3) and (7,7) position. So, the shortest path between the source and the destination is 10 hops length. We modify the existing destination sequenced distance



ω	γ	$\alpha = \omega + (1-\omega)\gamma$	PRRP HC	CRP HC
0	0	0	29	29
0	0.5	0.5	64.56	47.6
0	1	1	5435.81	159.86
0.5	0	0.5	65.46	64.37
0.5	0.5	0.75	149.22	101.57
0.5	1	1	6173.73	300.8
0.8	0	0.8	184.83	189.17
0.8	0.5	0.9	448.5	315.35
0.8	1	1	4703.54	719.77

Figure 5 Performance comparison betweeen PRRP and CRP.

vector (DSDV) routing protocol to adopt the PRRP and CRP. We run simulation for 3000 seconds and record the average hop count and delay for different γ and α keeping $\omega = 0$. Figure 5(g) and Figure 5(h) show the comparison between PRRP and CRP in terms of hop count and time. We observe similarity α or γ between $[0, 0.6]$ in hop count with our Java simulation but difference in delay. The delay is not directly proportional to the hop count as we assume in our java simulation. Though delay is not directly related to hop count we found less delay in the CRP compared to the PRRP. The delay depends on many factor including number of messages in a queue, wireless link status, number of re-transmission of a packet, nodes' capacity and transmission speed.

The hop count $\alpha or \gamma$ between $[0.6, 0.8]$ in java simulation increased exponentially but increased almost linearly in NS3 simulation. This is because the java simulation has huge number of nodes and higher average number neighbors than NS3 simulation. So, the packet can spread out to farther nodes resulting in higher hop counts. In NS3 simulation we use small number of (10x10) nodes. Therefore with higher probability of random routing could not spread out the packets resulting in a almost linear increment in hop count. Therefore, from the NS3 simulation we also conclude that the CRP has better performance over PRRP.

We try to conduct experiments in most possible ways to compare PRRP and CRP. We compare the performances in terms of average hop count and average queuing time of PRRP and CRP by varying different control variables. We compare the performances by changing the distance between the source and the destination. We also compare the performances in different node distributions and transmission range of nodes. Our simulation shows PRRP has a very large hop count when the distance is large. Therefore, from the above experiments it is clear that CRP performs much better than PRRP.

8 Conclusion

Source and destination location privacy is an important security property in wireless sensor networks. A secured routing protocol should be designed in such a way that the adversary will not be able to analyze the source or destination locations. There is a trade-off between security and efficiency in routing. If we want more security then we have to compromise efficiency and vice versa. In this paper, we proposed a new routing protocol to provide the source and the destination location privacy. Our proposed algorithm maintains a good balance of security and efficiency. Nodes that are far away from the source and the destination uses more efficient routing strategies and closer nodes uses more secured strategies. Our simulation results also show a great improvement in average hop count

Acknowledgement

This research was supported in part by NSF grants CNS 1629746, CNS 1564128, CNS 1449860, CNS 1461932, CNS 1460971, CNS 1439672, and ARO grant W911NF-17-1-0378.

References

[1] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms.

Commun. ACM, 24(2):84–90, 1981.

[2] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *J. Cryptol.*, 1(1):65–75, 1988.

[3] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding routing information. In *First International Workshop on Information Hiding*, pages 137–150, 1996.

[4] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *13th Conference on USENIX Security Symposium*, pages 21–21, 2004.

[5] Michael J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In *9th ACM Conference on Computer and Communications Security*, pages 193–206, 2002.

[6] Parisa Tabriz and Nikita Borisov. Breaking the collusion detection mechanism of morphmix. In *6th International Conference on Privacy Enhancing Technologies*, pages 368–383, 2006.

[7] Amos Beimel and SHLOMI Dolev. Buses for anonymous message delivery. *Journal of Cryptology*, 16(1):25–39, 2003.

[8] M. Conti, J. Willemsen, and B. Crispo. Providing source location privacy in wireless sensor networks: A survey. *IEEE Communications Surveys Tutorials*, 15(3):1238–1280, Third 2013.

[9] P. K. Roy, Rimjhim, J. P. Singh, and P. Kumar. An efficient privacy preserving protocol for source location privacy in wireless sensor networks. In *International Conference on Wireless Communications, Signal Processing and Networking*, pages 1093–1097, 2016.

[10] P. Kumar, J. P. Singh, P. Vishnoi, and M. P. Singh. Source location privacy using multiple-phantom nodes in wsn. In *IEEE Region 10 Conference (TENCON)*, pages 1–6, 2015.

[11] J. Celestine, K. Vallepalli, T. Vinayaraj, J. Almotir, and A. Abuzneid. An energy efficient flooding protocol for enhanced security in wireless sensor networks. In *Long Island Systems, Applications and Technology*, pages 1–6, 2015.

[12] Pandurang Kamat, Yanyong Zhang, Wade Trappe, and Celal Ozturk. Enhancing source-location privacy in sensor network routing. In *25th IEEE International Conference on Distributed Computing Systems*, pages 599–608, 2005.

[13] Y. Li and J. Ren. Source-location privacy through dynamic routing in wireless sensor networks. pages 1–9, 2010.

[14] W. Tan, K. Xu, and D. Wang. An anti-tracking source-location privacy protection protocol in wsns based on path extension. *IEEE Internet of Things Journal*, 1(5):461–471, 2014.

[15] Balaso N. Nadre, Devdatt Jagdale. Safety time improving for source node in wsn. In *2015 International Conference on Applied and Theoretical Computing and Communication Technology*, pages 620–624, 2015.

[16] M. Dong, K. Ota, and A. Liu. Preserving source-location privacy through redundant fog loop for wireless sensor networks. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pages 1835–1842, 2015.

[17] L. Lightfoot, Y. Li, and J. Ren. STaR: design and quantitative measurement of source-location privacy for wireless sensor networks. *Security and Communication Networks*, 9(3):220–228, 2016.

[18] D. Tang, T. Li, J. Ren, and J. Wu. Cost-aware secure routing (CASER) protocol design for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(4):960–973, 2015.

[19] Pandurang Kamat, Yanyong Zhang, Wade Trappe, and Celal Ozturk. Enhancing source-location privacy in sensor network routing. In *IEEE 25th International Conference on Distributed Computing Systems*, pages 599–608, 2005.

[20] J. Chen, Z. Lin, and X. Du. Protecting sink location against global traffic monitoring attacker. In *2016 International Conference on Computing, Networking and Communications*, pages 1–5, 2016.

[21] Y. Li, J. Ren, and J. Wu. Quantitative measurement and design of source-location privacy schemes for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(7):1302–1311, 2012.