# Fast mobility control protocols with sink location protection in wireless sensor networks

Xiao Chen* and Qijun Gu

Department of Computer Science,
Texas State University-San Marcos,
San Marcos, TX 78666, USA
E-mail: xc10@txstate.edu
E-mail: qg11@txstate.edu
*Corresponding author

Zhen Jiang

Department of Computer Science,
West Chester University,
West Chester, PA 19383, USA
E-mail: zjiang@wcupa.edu

Jie Wu

Department of Computer and Information Sciences,
Temple University,
Philadelphia, PA 19122, USA
E-mail: jiewu@temple.edu

**Abstract:** In this paper, we propose several novel fast converging mobility control protocols to achieve optimal deployment for streaming data flows in wireless sensor networks (WSNs). We also identify a unique sink privacy issue in mobility control protocols and propose a new privacy-preserving scheme which can be integrated into mobility control protocols to protect the sink. The simulation and performance analysis show that the proposed privacy-preserving mobility control protocols, with the sink node well-protected, can reach near-optimal results in terms of convergence speed, energy consumption and the communication cost.

**Keywords:** anonymity; mobility control; privacy-preserving; sink location protection; wireless sensor networks; WSNs.

**Reference** to this paper should be made as follows: Chen, X., Gu, Q., Jiang, Z. and Wu, J. (2011) 'Fast mobility control protocols with sink location protection in wireless sensor networks', *Int. J. Internet Protocol Technology*, Vol. 6, Nos. 1/2, pp.3–17.

**Biographical notes:** Xiao Chen is an Associate Professor of Computer Science at Texas State University-San Marcos. She received her PhD in Computer Engineering from Florida Atlantic University in 1999, Masters and Bachelors in Computer Science from Shanghai University of Science and Technology (now Shanghai University) in 1995 and 1992. Her research interests are in the areas of delay tolerant networks, sensor networks and ad hoc wireless networks.

Qijun Gu is an Assistant Professor of Computer Science at Texas State University-San Marcos. He received his PhD in Information Sciences and Technology from Pennsylvania State University in 2005, Masters and Bachelors from Peking University, China, in 2001 and 1998. His research interests cover various topics on network, security and telecommunication. He has been working on projects including denial of service in wireless networks, key management in broadcast services, worm propagation and containment, genetic algorithm in network optimisation, etc. His current projects include vulnerability in sensor applications, security in multichannel wireless networks, authentication in ad hoc and sensor networks, and security in peer to peer systems.

Zhen Jiang received his BS from Shanghai Jiaotong University, China, in 1992, Masters from Nanjing University, China, in 1998, and PhD from Florida Atlantic University, USA, in 2002. Currently, he is an Associate Professor of Computer Science Department at West Chester University of Pennsylvania, USA. His research interests are in the areas of information system development, routing protocols, and wireless networks. He is a member of the IEEE.

Jie Wu is a Distinguished Professor and Chair in the Department of Computer and Information Sciences at Temple University, and was the Program Director at US National Science Foundation. He has published over 450 papers in various journals and conference proceedings. His research interests are in the areas of wireless networks and mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. He served as the Chair of IEEE TCDP Committee. He also served as an Editor, Program Chair, and Co-chair of numerous international committees, conferences, and journals. He is the author of the text *Distributed System Design* published by the CRC press. He is an IEEE Fellow.

## 1 Introduction

With the advance in technology, wireless sensor networks (WSNs) performing sensing and communication tasks will be widely deployed in the near future because they greatly extend our ability to monitor and control the physical environment and improve the accuracy of our information gathering (Estrin et al., 2002; Hu et al., 2003; Kahn et al., 2000; Reif and Wang, 1999; Sibley et al., 2002). Sensor nodes can be deployed in inhospitable physical environment such as battlefields, remote geographic regions, and toxic urban locations. And now, sensors can be tiny and mobile as they can be a group of mobile robotic insects sensing dangerous areas or enemy targets and sending back as much information as possible (Wood, 2007).

In WSNs, power in sensors is the scarcest resource. In many situations, sensors cannot get recharged very often after being deployed in the field. Therefore, power efficient communication mechanisms are desired. As mobility becomes readily available to sensors (Rodoplu and Meng, 1999), recent studies on using mobility as a control mechanism to minimise energy consumption (Goldenberg et al., 2004; Jiang et al., 2007; Li and Halpern, 2001; Wang et al., 2005, 2007) have been conducted. Several mobility control protocols have been developed in which mobile sensors are controlled to move to the most power efficient positions for communication. These studies show that the saved energy in communication can compensate for the energy consumption in movement. Thereby, the overall energy consumption of sensors is reduced. These protocols also ensure that the communication among sensors will not be disrupted when sensors are moving to their best locations.

However, we observe that the existing protocols converge slowly, i.e., it takes many rounds for the sensors to move to their optimal locations because a sensor node adjusts its position according to the positions of its two neighbours. Thus, a lot of energy is wasted in movement. To address this problem, we propose two new mobility control protocols that can converge quickly while keeping all the advantages of the existing protocols. The mobility control protocol with quick convergence (MCC) speeds up the convergence process by avoiding the overreaction of a node to the movement of its neighbours, while the mobility control protocol with fast convergence (MCF) reduces the convergence time by moving the nodes as close to their optimal positions as possible. The key idea of our protocols is to use the optimal location information of each relay node calculated by algorithm mobility control with minimum total moving distance (MCM) as a guide for mobility control. Experimental results show that this information allows the mobility control process to converge quickly. The idea of MCM is put forward in Goldenberg et al. (2004). It is a lightweight algorithm that can be carried out along with any routing protocol by simply appending a counter and the location information of the source and the destination to the routing messages.

Furthermore, the existing mobility control protocols do not consider any security issue. In particular, the security of the sink node has never been discussed. The sink node in a WSN is crucial for gathering, aggregating, transferring and processing sensor information. If a sink node is located and destroyed, the network covered by the destroyed sink node will not function. Although we notice that some existing security schemes for WSNs can be used to prevent packets from being eavesdropped or modified (Karlof et al., 2004; Drissi and Gu, 2006) and to secure credentials used in mobility control (Eschenauer and Gligor, 2002; Zhu et al., 2003), the current mobility control protocols are risk free for attackers to obtain the sink location information, because mobility control protocols yield the location information of the sink node to mobile sensors. Attackers can compromise some nearby nodes to obtain sink location. Such attacks can be launched by attackers anywhere, even far away, from the sink node without exposing the attackers themselves. Therefore, protecting the location of the sink node is one of the critical security issues to safeguard WSN operations. Thereby, we identify this issue as *sink location privacy* which is defined as hiding the location of the sink node. However, the sink location information can be hardly protected using existing security mechanisms. At the same time, a scheme for sink protection should not affect normal sensing, communication and mobility control tasks that require the knowledge of the sink location. To address this privacy issue, we propose a novel privacy-preserving scheme, called *sink-anonymity scheme* (SAS) that only discloses the location information of dummy sinks and hides the real sink in a Φ-anonymity area to deceive attackers.

The contributions of the paper are four-fold. First, our mobility control protocols MCC and MCF converge much faster and reach nearly optimal results, compared with the existing mobility control protocols. Second, our protocols are fully distributed in that they use only one hop neighbourhood information, and they are proved to maintain the connections between a node and its neighbours. Third, the privacy of the sink location is a unique issue in mobility control. It has not been given much attention in the sensor network research field. Most security and privacy related research focuses on secure routing, key management, source privacy and denial of service. Fourth, the SAS is the first work to address the sink location privacy issue in mobility control. The scheme does not disclose the sink location, or any information to help attackers derive the sink location. We show that it has Φ-anonymity on the sink location and can be readily integrated into existing mobility control protocols to enhance their security.

The rest of the paper is organised as follows. Section 2 summarises existing mobility control protocols and related privacy issues in WSNs. Section 3 provides the preliminary information on mobility control. Section 4 presents two novel mobility control protocols with fast convergence. Section 5 describes the privacy-preserving scheme and proves its Φ-anonymity on the sink location. Section 6 shows how to apply the privacy scheme in the current mobility control protocols. Section 7 presents the results of simulation and analysis on performance. Finally, Section 8 concludes the paper.

## 2 Related works

### *2.1 Mobility control*

Power-efficient topology control and routing protocols (Perkins, 2000; Royer and Toh, 1999) have been well studied in the past years. A few recent studies (Goldenberg et al., 2004; Jiang et al., 2007; Li and Halpern, 2001; Wang et al., 2005, 2007) have showed the feasibility of using mobility as a control primitive to minimise power/energy consumption in networks of mobile sensors. In Goldenberg et al. (2004), the authors prove that in a single active flow between a source and a destination pair, if the energy cost function is a non-decreasing convex function, the optimal positions of the relay nodes must lie entirely on the line between the source and destination, and that the relay nodes must be evenly spaced along the line. Based on this, despite the randomness of the initial deployment, if nodes can move toward their optimal locations under mobility control, the energy consumption in communication can be minimised.

Following the work, a few mobility control protocols have been proposed and implemented. Synchronous and asynchronous mobility control algorithms (Goldenberg et al., 2004) let relay nodes reach their optimal locations based on the averaging algorithm (Jadbabaie et al., 2003; Rao et al., 2003). In brief, each node's optimal location is the average of its *left* and *right* neighbours' locations. The left and right neighbours of a node refer to the left and right neighbours on the line between the source and the destination. Thus, a node moves along with the movement of its two neighbours. The algorithms are simple: they only require one hop local information to be exchanged between a node and its left and right neighbours, and are distributed, which is suitable for a mobile environment. Also, the authors prove that the movement of a node in this way will not break the connections between the node and its neighbours. However, Jiang et al. (2007) find a problem in the algorithms: nodes may oscillate around their optimal locations and deplete their energy. A solution using a predefined threshold $\delta$ is provided (Jiang et al., 2007) to address the problem so that nodes will stop moving when the distance between the node's current location and the next location is no greater than the threshold.

As discussed before, these protocols do not make nodes move quickly to their optimal locations. Instead, many rounds of movement are needed for them to reach their final locations. Such slow convergence is a negative factor to justify the effectiveness of the mobility control primitive in power-efficient communication. Our protocols will address this critical issue.

### *2.2 Privacy*

Privacy research was mainly conducted in the context of information privacy and anonymisation. For example, a packet or a traffic pattern should not disclose identity information. Anonymous communication is one of the main research topics (Reed et al., 1998; Chaum, 1981). In this type of communication, a series of intermediate systems (mixes) are deployed. Each mix accepts messages from multiple sources, performs one or more transformations on them, and then forwards them in a random order. This method can hide the source and destination addresses in IP routing when being deployed in the internet. However, it is not suitable in mobility control, because location information is not identity information.

A few schemes (Xi et al., 2006; Mehta et al., 2007; Kamat et al., 2005) have been proposed on *source location privacy* in sensor networks. Their defence objective is to protect the object that is being monitored by sensors. Because the object being monitored is usually around the source nodes that are sending information back, attackers can locate the object by locating the source nodes. Hence, their schemes are focused on how to hide the locations of the source nodes. The main ideas of these schemes can be summarised as follows:

1 each source node floods packets through numerous paths to the base station to make it difficult for an adversary to trace the source

2 each real source node is associated with a few other (real or fake) source nodes so that they all generate packets at the same time to confuse attackers

3 a source node sends a packet in a looping path that goes through the base station so that attackers will get lost in it

4 all source nodes periodically send back packets regardless of whether they are monitoring the object or not

5 a set of virtual objects are put in the field to simulate the behaviour of the real object so as to hide it.

In this paper, we are interested in sink location privacy. Because the sink node in a sensor network is crucial for gathering, aggregating and transferring sensor information, if the location of the sink node is disclosed and the sink is destroyed, the functionality of a sensor network can be sabotaged. The problem of sink location privacy in mobility control is apparently different from the source location privacy in that the sink node is usually the destination of routes in WSNs. We cannot simply use fake sink nodes distributed in the network to hide the real sink location, because all packets need to reach the real sink node. Even if fake sink nodes can forward packets to the real sink node, attackers can still try to locate the sink node by locating the fake sink nodes first. Hence, new schemes are needed to ensure the privacy of the sink node in mobility control protocols.

## 3 Preliminary

In this section, we introduce the background of mobility control protocols.

### 3.1 Assumptions

We assume that all sensor nodes have the same transmission range $R$. If two sensor nodes are within each other's transmission range, they can communicate directly and they are called neighbours. Otherwise, they have to rely on intermediate nodes to relay messages for them. We define a WSN as a graph $G = (V, E)$, where $V$ is the set of all sensor nodes and $E$ is the set of all edges between pairs of sensor nodes. If two sensor nodes can communicate directly, there is an edge between them in $G$. The location of each node $u$ is $(x_u, y_u)$, simply denoted as $L(u)$. $|L(u) - L(v)|$ is the physical distance between two nodes $u$ and $v$. $L'(u)$ denotes the target location of $u$ in its movement and $L^*(u)$ is the optimal location of $u$.

We assume neighbours can share their location information by exchanging short messages. Location information can be discovered by GPS or some GPS-free positioning algorithms such as the one in Capkun et al. (2001). To simplify the discussion, we describe the protocols in a synchronous, round-based system, where each mobility control message is sent and received in the same round. All the protocols presented in the paper can be extended to an asynchronous system. However, to make our protocols clear, we do not pursue the relaxation.

We assume that a path from the source $s$ to the destination $d$ has already been discovered using a routing protocol, e.g., a greedy routing protocol or one of the ad hoc routing protocols. We also assume that both $s$ and $d$ are not moving during the process. Otherwise, the path is always broken and a new routing path needs to be established.

### 3.2 Mobility control

Denote the location of the source $s$ as $L(s)$, the location of the destination $d$ as $L(d)$, and the intermediate relay nodes as $u_i$ for $i \in [1, n-1]$. Accordingly, $u_0$ is source $s$ and $u_n$ is destination $d$. According to Goldenberg et al. (2004), the optimal location $L^*(u_i)$ of $u_i$ can be calculated as

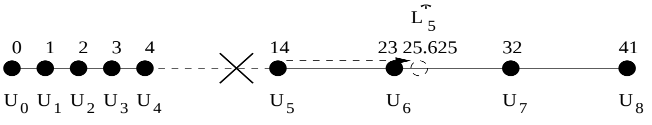
$$L^*\left(u_i\right) = L(s) + i \times \frac{L(d) - L(s)}{n} \tag{1}$$

**Algorithm MCM** (Goldenberg et al., 2004): Mobility control with minimum total moving distance.

```
1: The source node $s$ sends $L(s)$ and its label 0 to $u_1$. When
   each relay node $u_i$ receives $L(s)$ and the label $i - 1$, it will
   pass $L(s)$ and its own label $i$ to the succeeding node along
   the path. Such a propagation will end at $d$.
2: Once $L(s)$ is received at the destination node $d$, $d$ sends a
   message carrying $L(d)$ back to $s$ along the path.
3: At each relay node $u_i$, once both $L(s)$ and $L(d)$ are
   received, set $L^*\left(u_i\right) = L(s) + i \times \frac{L(d) - L(s)}{n}$ and move $u_i$
   to $L^*(u_i)$.
```

The details of the algorithm are presented in Algorithm MCM here. Algorithm MCM has a very nice property: the total moving distance of nodes in MCM is minimum because each node can move to its optimal location in one hop. However, this can create a problem: the movement can break the connections between a node $u_i$ and its neighbours. For example, suppose nine nodes $u_0, u_1, \cdots, u_8$ are aligned in a line (see Figure 1). Node $u_0$ is the source and $u_8$ is the destination. The transmission range of each node is 10. Node $u_i$ ($i \in [0..4]$) is at location $i$. Node $u_5$ is at location 14, node $u_6$ is at 23, node $u_7$ is at 32 and node $u_8$ is at location 41. According to Algorithm MCM, the optimal location of relay node $u_i$ should be $L^*(u_i) = 0 + (41 - 0) = 8 * i$. Therefore, the optimal location of node $u_5$ (denoted as $L_5^*$) is 25.625. If node $u_4$ is still at its old location 4 when node $u_5$ moves to its optimal location, the connection between them is lost. When neighbours are disconnected, the data sent is lost and has to be resent from the source after timeout. The neighbours have to reconnect through sending each other Hello messages. As indicated in Cortes et al. (2004), Mao and Wu (2005), and Poduri and Sukhatme (2004), all of these will decrease the communication efficiency.

**Figure 1** Broken link between $u_4$ and $u_5$ in MCM



To address the disconnection problem, a distributed algorithm that allows the relay nodes to move to their optimal positions is introduced as shown in Algorithm MCD (Goldenberg et al., 2004). The key ingredient of this algorithm is the simple average calculation. A relay node always only moves toward the average of its two neighbours, instead of reaching its optimal location in one round. Value $g$ in the algorithm is a damping factor that prevents a node from overacting to the movement of its two neighbours. It is proved (Goldenberg et al., 2004) that the connection between communicating neighbours using MCD will not be broken.

**Algorithm MCD** (Goldenberg et al., 2004): Mobility control at each relay node $u_i$.

```
Exchange $L(u_i)$ with $u_{i-1}$ and $u_{i+1}$.
Receive $L(u_{i-1})$ and $L(u_{i+1})$. Set
    $L'(u_i) = \frac{L(u_{i-1}) + L(u_{i+1})}{2}$.
Set damping factor $g$ a random value $\in (0, 1]$, move
    toward $L(u_i) + g \times (L'(u_i) - L(u_i))$.
```

## 4 Fast convergence mobility control protocols

In this section, we first describe our motivation to develop fast convergence mobility control protocols by pointing out the problems of MCD and then propose two mobility control protocols, MCC and MCF, to let nodes move to their optimal locations much more quickly.

### *4.1 Oscillation and slow convergence of MCD*

Although Algorithm MCD can make nodes move to their optimal locations, it suffers from oscillation and slow convergence problems as explained in the following example.

In Table 1, there are six nodes, including the source and the destination. The transmission range of each node is 10. Suppose they are placed in a line and the $y$ coordinate of each node is the same. Therefore, in the table, only the $x$ coordinate of each node is shown. Round 0 displays the initial location of each node. Starting from round 1, each relay node uses MCD (suppose $g$ is set to 1) to calculate its optimal location. From the table, we can see that the process for each node to reach its optimal location converges slowly. It takes the nodes 53 rounds to reach their close-to-optimal locations. Then, the nodes start to oscillate around their close-to-optimal locations and never stop. This kind of oscillation is caused by the round-off errors in computers. It wastes computation resources and will deplete the energy in nodes very quickly.

The oscillation problem can be easily solved by setting a threshold $\delta$ (Jiang et al., 2007) such that if the difference between node $u_i$'s target position $L(u_i)$ and the current position $L(u_i)$ is less than $\delta$, the node does not have to move any more.

To tackle the slow convergence problem without loosing the connectivity between communicating neighbours, we propose two mobility control protocols MCC and MCF. Both protocols use the optimal locations of the relay nodes calculated by MCM as a guide for nodes movement.

### *4.2 Protocol MCC*

The first protocol MCC combines the ideas of MCM and MCD (details shown in Algorithm MCC). This protocol still uses the average calculation in MCD. The difference is that in MCD, a node will move as the locations of its left and right neighbours change. However, in MCC, a relay node knows its optimal position by MCM, and if the distance between its new position (which is calculated as the average of its two neighbours' positions) and its optimal position is larger than the distance between its current position and its optimal position, it does not move. In this way, a node can avoid unnecessary movement. Therefore, the time it takes to complete the convergence process can be reduced.

Note that the MCM part of the algorithm only needs to be called once if the locations of the source, the destination, and the label of each relay node do not change for a period of time. It is used here to calculate the optimal locations of relay nodes, not making them move immediately to those locations. So, it will not cause the disconnection problem. The actual movement is done in the loop part of MCC. The calling of MCM does not create much overhead because it can be incorporated into the routing process. When source $s$ sends a message to $d$, it can also send its $L(s)$ and its label 0 along with the message. Each intermediate node will do the same thing until the message reaches $d$. Then $d$ sends an acknowledgment plus its $L(d)$ back to $s$. When each relay node $u_i$ has $L(s)$, $L(d)$ and its label, it can calculate its optimal position. After nodes reach their optimal locations, according to Goldenberg et al. (2004), the subsequent routing energy consumption can be minimised. It can be proved that the connectivity of communicating neighbours is kept in MCC.

**Algorithm MCC:** MCM combined with MCD.

```
Apply MCM to obtain the optimal location $L^*(u_i)$ for each
    intermediate node $u_i$.
repeat
    for each intermediate node $u_i$ at round $t$ do
        Exchange the location $L_{t-1}(u_i)$ reached in the last
            round $t-1$ with $u_{i-1}$ and $u_{i+1}$.
        Receive $L_{t-1}(u_{i-1})$ and $L_{t-1}(u_{i+1})$. Set
            $L_t(u_i) = \frac{L_{t-1}(u_{i-1}) + L_{t-1}(u_{i+1})}{2}$.
        If $|L_t(ui) - L^*(u_i)| < |L_{t-1}(u_i) - L^*(u_i)|$ and
            $|L_t(u_i) - L^*(u_i)| > \delta$, then move to $L_t(u_i)$.
    end for
until all nodes have no further movement.
```

**Figure 2** Illustration of Theorem 1

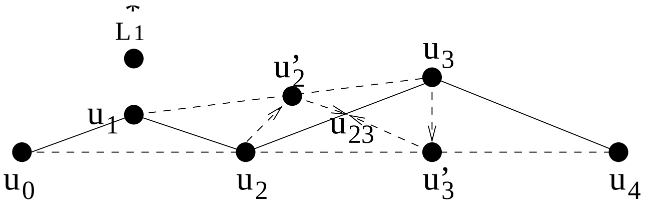


**Table 1** The slow convergence process and oscillation of nodes to reach their optimal locations

| *Round* | $s_x$ | $node1_x$ | $node2_x$ | $node3_x$ | $node4_x$ | $d_x$ |
|---|---|---|---|---|---|---|
| 0 | 92.11134 | 86.99914 | 80.11193 | 74.99975 | 69.11155 | 63.99937 |
| 1 | 92.11134 | 86.11163 | 80.99944 | 74.61174 | 69.49956 | 63.99937 |
| 2 | 92.11134 | 86.55539 | 80.36169 | 75.24949 | 69.30556 | 63.99937 |
| 3 | 92.11134 | 86.23651 | 80.90244 | 74.83362 | 69.62444 | 63.99937 |
| .. | .. | .. | .. | .. | .. | .. |
| 53 | 92.11134 | 86.48894 | 80.86656 | 75.24416 | 69.62177 | 63.99937 |
| 54 | 92.11134 | 86.48895 | 80.86655 | 74.24417 | 69.62176 | 63.99937 |
| 55 | 92.11134 | 86.48894 | 80.86656 | 74.24416 | 69.62177 | 63.99937 |
| 56 | 92.11134 | 86.48895 | 80.86655 | 74.24417 | 69.62176 | 63.99937 |
| 57 | 92.11134 | 86.48894 | 80.86656 | 74.24416 | 69.62177 | 63.99937 |
| 58 | 92.11134 | 86.48895 | 80.86655 | 74.24417 | 69.62176 | 63.99937 |
| .. | .. | .. | .. | .. | .. | .. |

*Theorem 1:* Connectivity is kept between communicating neighbours in MCC.

*Proof:* Without loss of generality, in our proof, we need to cover cases where a node will move with the location changes of its two neighbours and cases where a node will not move if the new location is farther away from its optimal location than its current location. To cover both cases, we come up with a network as shown in Figure 2. There are five relay nodes $u_0$, $u_1$, ⋯, $u_4$. For convenience's sake, the labels of the nodes are also used for their locations. A solid line between two nodes indicates that they can communicate with each other directly. An undirected dashed line is used to indicate the distance between them. And a directed line represents the movement of a node.

MCM calculates the optimal locations of the relay nodes. In the figure, only the optimal location $(L_1^*)$ of node $u_1$ is shown. Node $u_1$ is the one that will not move because its new location which is at the midpoint of $u_0$ and $u_2$ is farther away from its optimal location. All others will move to their new locations, that is, node $u_2$ will move to $u_2'$ which is the midpoint of $u_1$ and $u_3$, and node $u_3$ to $u_3'$ which is at the middle of $u_2$ and $u_4$.

Now, we want to prove that the connections of nodes in their new locations are not lost. That is, $|u_1 - u_2'|$ and $|u_2' - u_3'|$ are less or equal to the transmission range $R$.

First, we prove that $|u_1 - u_2'| \le R$ is true. Obviously in triangle $u_1u_2u_3$, either $|u_1 - u_2| \ge |u_1 - u_2'|$ is true or $|u_2 - u_3| \ge |u_2' - u_3|$ is true. Since $|u_1 - u_2'| = |u_2' - u_3|$, $|u_1 - u_2| \le R$, and $|u_2 - u_3| \le R$ are true, $|u_1 - u_2'| \le R$ is also true.

Next, we prove that $|u_2' - u_3'| \le R$ is true. Denote the midpoint of $u_2u_3$ as $u_{23}$. $|u_2' - u_3'| \le |u_2' - u_{23}| + |u_3' - u_{23}| = \frac{1}{2}(|u_1 - u_2| + |u_3 - u_4|) \le \frac{1}{2}(R + R) = R.$ So $|u_2' - u_3'| \le R$ is true.

Therefore, the connectivity is not lost in Algorithm MCC. □

### 4.3 Protocol MCF

The second protocol MCF also uses MCM to obtain the optimal location for each relay node. The idea is that the relay nodes should move toward their optimal locations as much as possible without breaking the connections with their left and right neighbours. In this way, for each node, there is no extra movement. The details of this algorithm are shown in Algorithm MCF.

**Algorithm MCF:** Move to optimal location as much as possible.

```
Apply MCM to obtain the optimal location $L^*(u_i)$ for each intermediate node $u_i$.
repeat
    for each intermediate node $u_i$ at round $t$ do
        Calculate target location $L_t(u_i)$ which is the closest point to $L^*(u_i)$ without breaking the connection with $u_i$'s left and right neighbours $u_{i-1}$ and $u_{i+1}$.
        If $|L_t(u_i) - L^*(u_i)| > \delta$, then move to $L_t(u_i)$.
    end for
until all nodes have no further movement.
```

In MCF, the target location $L_t(u_i)$ in each round $t$ can be easily calculated using a small program that solves mathematical equations. Theorem 2 shows that the connection between communicating neighbours is not lost in MCF.

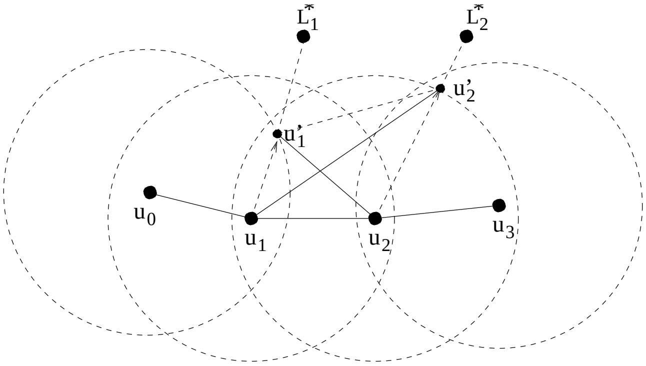
*Theorem 2:* Connectivity is kept between communicating neighbours in MCF.

*Proof:* Without loss of generality, suppose there are four relay nodes $u_0$, $u_1$, $u_2$, $u_3$ (see Figure 3). The area covered by a node's transmission range is represented by a dashed circle in the figure. A solid line between two nodes indicates that they can communicate with each other directly. An undirected dashed line is used to indicate the distance between them. And a directed line represents the movement of a node. Node $u_1$ has neighbours $u_0$ and $u_2$ while node $u_2$ has neighbours $u_1$ and $u_3$. The optimal locations of nodes $u_1$

and $u_2$ are $L_1^*$ and $L_2^*$ respectively from MCM. Now, node $u_1$ and node $u_2$ will move toward their optimal locations as much as possible without loosing the contact with their neighbours. The new locations of nodes $u_1$ and $u_2$ are $u_1'$ and $u_2'$ as shown in Figure 3.

Now we want to show that the communication between nodes $u_1$ and $u_2$ in their new locations is not lost, that is, $|u_1' - u_2'| \leq R$. In Figure 3, in shape $u_1 u_2 u_2' u_1'$, either $|u_1' - u_2'| \leq |u_1' - u_2|$ is true or $|u_1' - u_2'| \leq |u_1 - u_2'|$ is true. Since $|u_1 - u_2'| \leq R$ and $|u_1' - u_2| \leq R$ are true, so $|u_1' - u_2'| \leq R$ is true. This means that the communication between nodes $u_1$ and $u_2$ in their new locations is still within the range $R$. □

**Figure 3** Illustration of Theorem 2



## 5 Privacy preserving

In this section, we introduce a unique security issue called sink location privacy in the proposed mobility control protocols, and put forward a privacy preserving scheme.

### *5.1 Attacks on sink location*

Mobility control protocols are susceptible to various attacks. Many security schemes of authentication, encryption and key management proposed in the past can be deployed in WSNs to protect them. But this paper will not address traditional attacks. Rather, we show two attacks (*direct attack* and *intersection attack*) that can be hardly defeated by current security countermeasures. The two attacks give attackers an easy access to sink location without exposing themselves. Using the two attacking approaches, attackers do not need to physically trace along a path hop by hop to the real sink node, but simply monitor nearby traffic or capture a few nearby nodes at any place far away from the real sink.

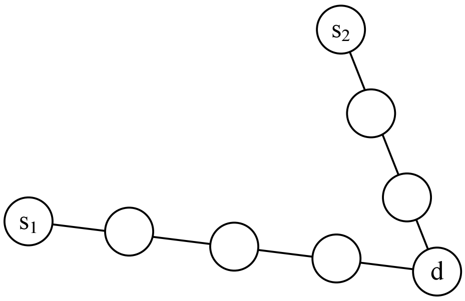
The direct attack exploits the fact that all mobility control protocols need to send the sink location information to the relay nodes. This is because MCC and MCF use MCM where the sink node $d$ includes its location $L(d)$ in the reply message when it sends acknowledgment back to the source node $s$. After the reply reaches $s$, all the intermediate nodes on the path will have the location information of $d$. Apparently, an attacker can obtain the location of the sink node $d$ by compromising any node along the path and then destroy the sink. Therefore, a privacy preserving approach is needed to hide the sink location.

The intersection attack exploits the geometric characteristic of paths formed in mobility control. Because the sink node places itself on the paths, attackers can infer the actual sink location by locating nearby nodes in two disjoint paths. This intersection attack is illustrated in Figure 4, in which the sink is communicating with two sources $s_1$ and $s_2$ via two disjoint paths. If an attacker can find any two nodes on each of the two disjoint paths, the attacker can obtain the two paths going through these nodes. Then, the intersection of the two paths discloses the location of the sink node. This attack allows attackers to obtain sink location without cracking any packet of mobility control protocols.

The challenge of defeating the two attacks and ensuring the privacy of sink location is that traditional security mechanisms cannot hide the sink location information. For example, encryption of the sink location cannot prevent a fully compromised node from disclosing the information because an attacker can easily obtain all credentials (such as keys) in the compromised node to decrypt any encrypted information.

Sink location anonymity is different from identity anonymity where attackers can eavesdrop packets transmitted near them and find the identity information of the sink node (such as its IP address). But this kind of information cannot help the attackers to locate the sink node. In this paper, we are concerned with the attacks that attempt to discover the location of the sink node and destroy it. Hence, the objective of our work is to hide the sink location.

**Figure 4** Intersection attack



### *5.2 Dummy node*

Our basic idea is to use a dummy node $d'$ to hide the real sink location information $L(d)$ from all the nodes on the path. Assume a sink node $d$ is communicating with several sources through multiple disjoint paths. After $d$ receives the source location $L(s)$ from each source, it does not reply with its real location $L(d)$. Instead, the sink node picks a one hop neighbouring node $h$ and a dummy node $d'$ for each path such that $h$ satisfies inequality (2) and $d'$ satisfies equation (3), where $n_x$ is the hop count of node $x$ from the source $s$ and $|L(x) - L(y)|$ is the distance between nodes $x$

and $y$. Inequality (2) states that $h$'s best location is in the communication range $R$ of $d$ so that $d$ is one hop away from the path that goes through $s$ and $h$. Equation (3) states that $d'$, $h$ and $s$ are on the same path.

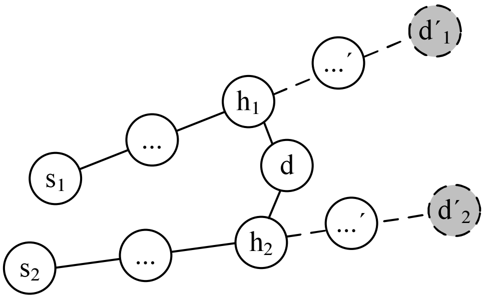
$$|L(h) - L(d)| < R \tag{2}$$

$$\frac{|L(d') - L(s)|}{n_{d'}} = \frac{|L(h) - L(s)|}{n_h} \tag{3}$$

The idea of dummy node is illustrated in Figure 5, where $d$ communicates with two sources $s_1$ and $s_2$ via its neighbours $h_1$ and $h_2$. Assume $d$ has obtained $s_1$'s location $L(s_1)$, $h_1$'s location $L(h_1)$, and the hop count $n_{h_1}$ between $s_1$ and $h_1$. To hide its true location from $s_1$, $d$ picks a random number $n_{d'_1} \gg n_{h_1}$ and computes the location $L(d'_1) = (L(h_1) - L(s_1))\frac{n_{d'_1}}{n_{h_1}} + L(s_1)$ according to equation (3). Then, $d$ claims a dummy sink $d'$ at $L(d'_1)$ on the path and $n_{d'_1}$ is the hop count between $s_1$ and $d'_1$.

Accordingly, $d$ can make two dummy nodes for the two disjoint paths as shown in Figure 5. $d$ selects $h_1$ and $d'_1$ for $s_1$, and $h_2$ and $d'_2$ for $s_2$. Although $d$ is not on either path, $d$ claims that it is the next hop to $h_i$ on the path. Hence, when $h_1$ receives a packet from $s_1$, it will forward the packet to its next hop which is $d$. $h_2$ will do the same thing after receiving a message. Thereby, $d$ can get information delivered in both paths while using the dummy nodes to counteract the direct and the intersection attacks on its location.

**Figure 5** Dummy nodes



*Property 1:* The dummy node does not disclose the real sink location if any mobile sensor node on a path is compromised.

*Proof:* Any sensor $u_i$ on a path between a source $s$ and a dummy sink $d'$ knows the locations of them. Compromising $u_i$, attackers can obtain the path equation as $y = \frac{Y_s - Y_{d'}}{X_s - X_{d'}}(x - X_s) + Y_s$, where $X_x$ and $Y_x$ are the coordinates of node $x$. Since the sink node is off the path, its location does not satisfy the path equation. Therefore, the path equation does not disclose the sink location. □

*Property 2:* The dummy node does not disclose the sink location if the intersection points of any two disjoint paths is compromised.

*Proof:* Assume that two disjoint paths intersect at point $x$. $x$ must satisfy the path equations of the two disjoint paths. Because the sink node is off both paths, the sink node does not satisfy the path equation of either path. Therefore, the intersection point $x$ does not disclose the sink location. □

### 5.3 *Φ-anonymity*

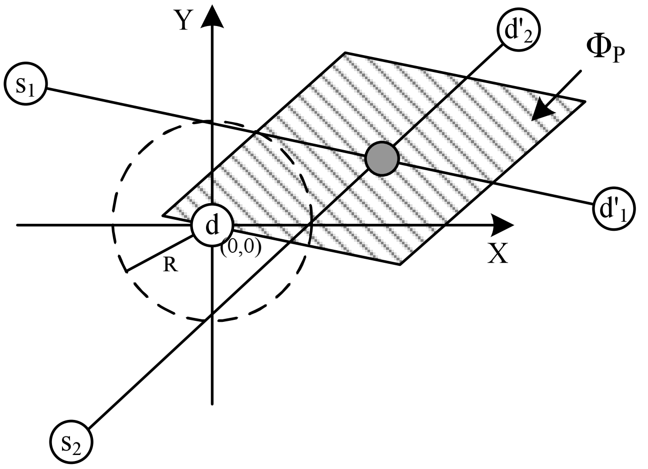
In this section, we propose a formal privacy model to analyse the achieved privacy. The model defines a *proximity area* surrounding the real sink node, which shows the range of the sink location that attackers can derive. The model can help the sink node to compare the privacy when different dummy nodes are selected.

As shown in Figure 6, we place the sink node at the origin. Assume that attackers have found $N$ disjoint paths. Each path equation $i$ is denoted as $y = k_i x + c_i$, for $1 \le i \le N$.

Because all these paths pass inside the communication range $R$ of the sink node, the vertical distance from the origin to any path is less than $R$. Hence, it must be true that $\frac{|c_i|}{\sqrt{1+k_i^2}} < R$.

Similarly, in order to determine whether the sink node is at the location $(X, Y)$ or not, attackers need to compute the vertical distance from the location $(X, Y)$ to each path $i$ using $d_i = \frac{|c_i + k_i X - Y|}{\sqrt{1+k_i^2}}$. If all $d_i$ satisfy inequality (4), i.e., the distance from the location $(X, Y)$ to any of the disjoint paths is less than the communication range $R$, a sink node might be at the location $(X, Y)$.

**Figure 6** Illustration of $\Phi_P$-anonymity



$$\frac{|c_i + k_i X - Y|}{\sqrt{1 + k_i^2}} < R \textit{ for } 1 \le i \le N \tag{4}$$

In the example of Figure 6, the proximity area is the shadowed area in which any position $(X, Y)$ satisfies inequality (4). Note that the proximity area covers the locations of possible sinks that are more than one hop away from the real sink node. As long as a location is within the communication range of either of the two disjoint paths, the location can be considered as the sink location. Hence, the proximity area is the achieved privacy against the two compromised paths.

Given such a proximity area, we define Φ-anonymity below. Accordingly, the proximity area in Figure 6 is a $\Phi_P$-anonymity, where $P$ is the set of the two disjoint paths.

*Definition 5.1:* Let Φ be a proximity area and $P$ be the set of all disjoint paths known to attackers. Φ is said to satisfy $\Phi_P$-anonymity if and only if Φ is the maximum proximity area in which any location $(X, Y)$ satisfies inequality (4) for all paths in $P$.

*Definition 5.2:* Let Φ be a proximity area and $P^*$ be the set of all disjoint paths known to the sink node. Φ is said to satisfy Φ-anonymity if and only if Φ is the maximum proximity area in which any location $(X, Y)$ satisfies inequality (4) for all paths in $P^*$.

The relation of $\Phi_P$-anonymity and Φ-anonymity is shown by Theorem 3 which indicates that attackers can reduce the proximity area if more disjoint paths are known. The smaller the proximity area is, the better estimation the attackers can have on the real sink location. However, the minimum proximity area that attackers can achieve is the Φ-anonymity area.

*Theorem 3:* A Φ-anonymity area is the minimum in all $\Phi_P$-anonymity areas, i.e., $\forall \Phi_P$, $\Phi \subseteq \Phi_P$.

*Proof:* Assume we can find a $P$ and a $\Phi_P$ such that $\Phi \nsubseteq \Phi_P$. Thereby, a location $x$ exists that $x \in \Phi$ but $x \notin \Phi_P$. Hence, the location $x$ is one hop away from all paths in $P^*$. However, a path $p \in P$ exists that $x$ is more than one hop away from $p$. Therefore, $p \notin P^*$ and thus $P \nsubseteq P^*$.

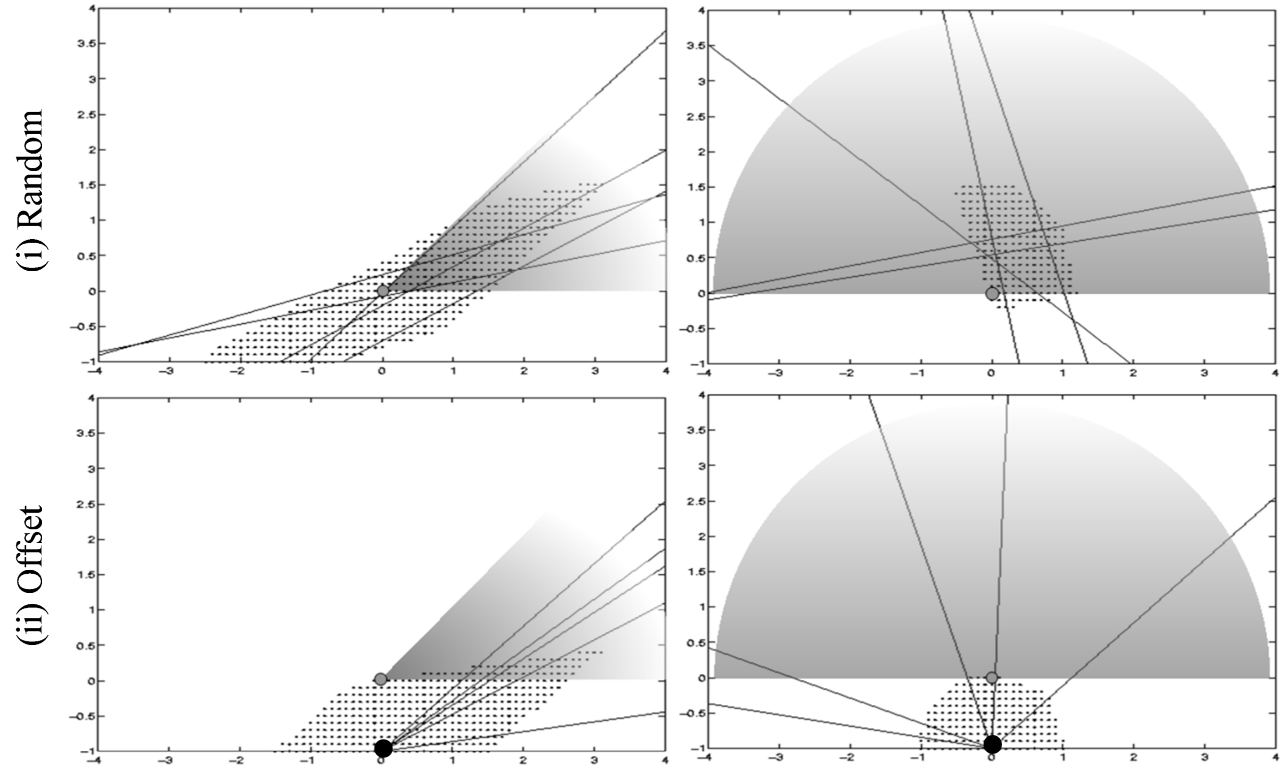
However, because $P$ is the set of all disjoint paths known to attackers and $P^*$ is the set of all disjoint paths known to the sink node, we know $P \subseteq P^*$, which contradicts $P \nsubseteq P^*$. Therefore, the theorem is proved by contradiction. □

## 5.4 Sink-anonymity scheme

The Φ-anonymity area is critical to the privacy of sink location given multiple paths. The larger the Φ-anonymity area is, the better the real sink node is protected. According to Definition 5.2, the shape and the size of the Φ-anonymity are determined by $P^*$. In other words, the disjoint paths selected by the sink node determine the privacy of the sink location.

We propose two SASs (shown in Algorithms R-SAS and O-SAS). R-SAS uses the *random disjoint path selection* approach, while O-SAS uses the *offset disjoint path selection* approach. By analysis and comparison, we show that only O-SAS can preserve privacy given multiple paths. And that is the one that will be embedded into our mobility control protocols later.

**Figure 7** Sink coverage and path selection approaches, (a) 45° (b) 180°



(a) (b)

To discuss the two schemes, we first model the area covered by a sink node as a ‘fan area’, i.e., all sensor nodes in the fan area report data to the sink node. Denote the angle of the fan as $\theta$. Figure 7 illustrates the fans of $\theta = 45°$ and $\theta = 180°$ (the grey areas). When $\theta = 180°$, paths to the sink node may come from all directions. Thereby, paths in a fan of $\theta > 180°$ are the same as paths in a fan of $\theta = 180°$. Note that the fan area of a sink node is normally determined by network deployment or task assignment. We assume the sink node has $\theta$ as a parameter in mobility control.

R-SAS makes each selected path go through a randomly positioned one hop neighbour and a randomly selected dummy sink. O-SAS does the same thing, and, in addition, makes all selected paths intersect at an *offset point x* (the black dots in the bottom row in Figure 7). Hence, in addition to equation (3) and inequality (2), all the paths selected by the offset approach also satisfy equation (5) which states that the offset point $x$ is on the paths. Note that $x$ is neither a dummy node nor a real node.

```
Algorithm R-SAS: Random sink-anonymity scheme.
for each requesting sensor $s$ do
    The sink node selects a neighbour $h$ and a dummy sink $d'$ such that
        (a) $h$ satisfies inequality (2),
        (b) $d'$ satisfies equation (3).
end for
```

```
Algorithm O-SAS: Offset sink-anonymity scheme
The sink picks an offset point $x$ in the offset area and keeps the offset from $x$ to $d$ as a secret.
for each requesting sensor $s$ do
    The sink node selects a neighbour $h$ and a dummy sink $d'$ such that
        (a) $h$ satisfies inequality (2) and equation (5),
        (b) $d'$ satisfies equation (3).
end for
```

$$\frac{Y_x - Y_s}{X_x - X_s} = \frac{Y_h - Y_s}{X_h - X_s} \tag{5}$$

Figure 7 shows examples of the $\Phi_P$-anonymity area (the dotted areas) when attackers know a set $P$ of five disjoint paths. The dotted areas illustrate several privacy properties. First, the actual sink node could be at any location within the dotted area. Knowing the area does not necessarily disclose the sink location. Second, the intersection of any paths does not disclose the sink location. When a sink node uses the offset selection method, the sink node can pick an offset point in any direction to hide itself. Hence, the offset point contributes no more information than the dotted area to attackers.

### 5.5 Privacy analysis of R-SAS and O-SAS

To analyse the privacy achieved by R-SAS and OSAS, we need to identify the Φ-anonymity areas in the proposed schemes. Theorem 4 shows that R-SAS does not provide any privacy protection to the real sink node if attackers compromise sensors in sufficient disjoint paths. On the contrary, Theorem 5 shows that O-SAS can achieve Φ-anonymity to protect the sink location.

*Theorem 4:* Given multiple disjoint paths, the Φ-anonymity area of R-SAS could be as small as the real sink node and thus reveal the real sink location.

*Proof:* Because the sink node randomly selects dummy nodes, it can possibly select two pairs of parallel paths $\{p_1, p_1'\}$ and $\{p_2, p_2'\}$ as shown in Figure 8(a). From there, let $P$ be the set of $\{p_1, p_1', p_2, p_2'\}$. Then, the sink node is the only location that is in one hop to all paths in $P$. Hence, ΦP includes only the sink node. Because $\Phi \subseteq \Phi_P$ as in Theorem 3, Φ includes only the sink node. □

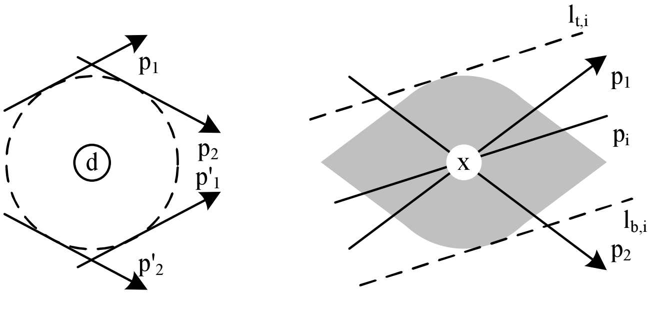
*Theorem 5:* Let $p_1$ and $p_2$ be the two outmost paths in O-SAS as shown in Figure 8(b) such that all paths in $P^*$ are bounded by the two paths. The Φ-anonymity area of O-SAS is the gray area in Figure 8(b) and does not reveal the real sink location.

*Proof:* For any $p_i \in P^*$, let $P_i = \{p_i\}$. Find two parallel lines $l_{t,i}$ and $l_{b,i}$ as in Figure 8(b) such that any point within the two lines is one hop away from $p_i$. Then, the area within the two lines is the $\Phi_{P_i}$-anonymity area.

We rotate $p_i$ from $p_1$ to $p_2$. For each instance of $p_i$, we find the corresponding $\Phi_{P_i}$-anonymity area. The overlapping area of all the $\Phi_{P_i}$-anonymity areas, which is the grey area in Figure 8(b), is the Φ-anonymity area.

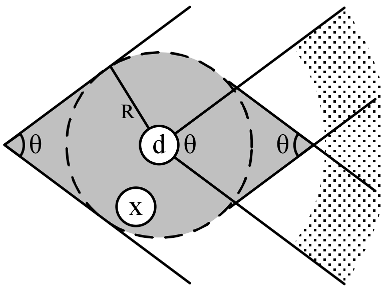
For O-SAS, the solid grey area in Figure 9 shows where the sink node can select an offset point $x$. The dotted area is where sensors are deployed. The farthest distance between the offset point and the sink node is $\frac{R}{\sin(\theta/2)}$. If $\theta \leq 60°$, the offset point could be more than two hops away from the real sink node, while any selected path passing the offset point is one hop away from the real sink node. □

**Figure 8** Analysis of Φ-anonymity, (a) R-SAS (b) O-SAS



(a) (b)

**Figure 9** Offset area



### *5.6 Comparison of R-SAS and O-SAS*

We use three metrics to quantitatively measure the Φ-anonymity of R-SAS and O-SAS: average distance $pe = \int_{\Phi} D_{(X,Y)} dXdY$, maximum distance $pm = \max_{(X,\ Y)\in\Phi}(D(X,\ Y))$ and area $p1 = \int_{\Phi} dXdY$, where $D(X, Y)$ is the distance of the location $(X, Y)$ to the real sink node.

**Algorithm SAMCC:** Sink-anonymity MCC.

```
The sink node $d$ picks a dummy sink $d'$ for each source $s$ according to O-SAS.
$d$ sends the location $L(d')$ back to $s$ via its neighbour $h$.
Apply MCM to obtain the optimal location $L^*(u_i)$ for each intermediate node $u_i$.
repeat
    for each intermediate node $u_i$ at round $t$ do
        Exchange the location $L_{t-1}(u_i)$ reached in the last round $t-1$ with $u_{i-1}$ and $u_{i+1}$.
        Receive $L_{t-1}(u_{i-1})$ and $L_{t-1}(u_{i+1})$. Set $L_t(u_i) = \frac{L_{t-1}(u_{i-1})+L_{t-1}(u_{i+1})}{2}$.
        If $|L_t(u_i) - L^*(u_i)| < |L_{t-1}(u_i) - L^*(u_i)|$ and $|L_t(u_i) - L^*(u_i)| > \delta$, then move to $L_t(u_i)$.
    end for
until all nodes have no further movement.
```

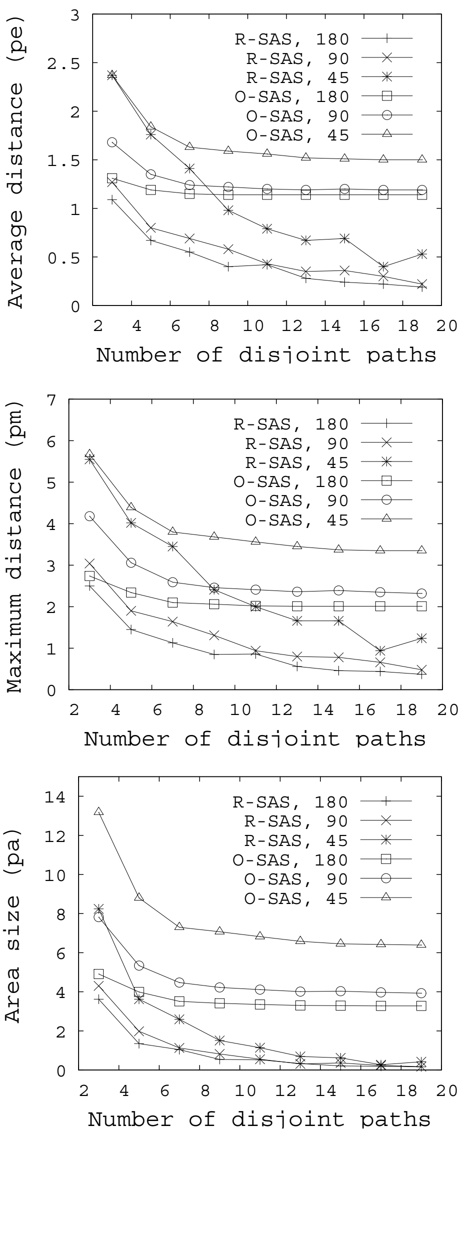
*pe* basically tells how far away the centre of the Φ-anonymity area is to the sink node in average. *pm* indicates the possible farthest location to the sink node. *pa* shows the size of the area where the sink node is. Therefore, from a defender's perspective, the larger the *pe*, *pm* and *pa*, the better the privacy.

Figure 10 depicts the measurement of privacy in three metrics. We study three situations, setting $\theta$ to 45°, 90° or 180°. We assume that the sink node has a few disjoint paths ranging from 3 to 19 in its covered area. The results are normalised as the communication range of the sink node is set to 1. All data points are averaged over 30 random scenarios generated in MATLAB.

First, the simulation confirms the privacy analysis of R-SAS and O-SAS. O-SAS provides much better privacy than R-SAS. As attackers obtain more disjoint paths, R-SAS in fact reduces the area where the sink node could be. For example, the area inferred from 19-disjoint paths in R-SAS is only about 4.5% of the area inferred from three disjoint paths. Thereby, attackers can estimate a very close location to the sink if they can find sufficient disjoint paths. In contrast, when O-SAS is used, the inferred area size reaches a boundary and cannot be further reduced as the number of disjoint paths increases. In other words, attackers cannot obtain the exact sink location by continuously trying more disjoint paths if O-SAS is applied.

Second, we observe that smaller $\theta$ implies better privacy to the sink node. When the sink node collects information from a smaller fan area, disjoint paths lay more parallel to each other. Their one hop surrounding areas thus have a larger overlap, which results in a larger Φ-anonymity area that attackers can infer. Thus, the sink node is better protected with a smaller $\theta$. This observation gives guidance to network deployment with mobility control. A sink node is better deployed at the boundary of a network than at the centre. A sink node is better assigned to monitor a part of the network than the whole network.

**Figure 10** Comparison of Φ-anonymity of R-SAS and O-SAS



## 6 Sink-anonymity mobility control protocols

In this section, we apply the O-SAS scheme to two mobility control protocols, MCC and MCF, and develop two new sink-anonymity mobility control protocols, SAMCC and SAMCF, that protect the real sink node and ensure the connectivity between communicating neighbours. We also apply the O-SAS scheme to MCM and MCD. They will be used in our simulation for comparison.

### *6.1 Protocol SAMCC*

In SAMCC, the sink node $d$ picks a dummy sink $d'$ according to O-SAS for the intermediate nodes to adjust their locations. An intermediate node knows its optimal position using MCM. If the distance between its new position (which is calculated as the average of its two neighbours' positions) and its optimal position is larger than the distance between its current position and its optimal position, it does not move. In this way, a node can avoid unnecessary movement.

For a particular source $s$, once a dummy node $d'$ is set, all the real intermediate nodes between the source and the dummy sink will move to their optimal locations. SAMCC will not disconnect communicating neighbours during the process. The complete algorithm is shown in Algorithm SAMCC.

**Algorithm SAMCF:** Sink-anonymity MCF.

```
The sink node $d$ picks a dummy sink $d'$ for each source $s$ according to O-SAS.
d sends the location $L(d')$ back to $s$ via its neighbour $h$.
Apply MCM to obtain the optimal location $L^*(u_i)$ for each intermediate node $u_i$.
repeat
    for each intermediate node $u_i$ at round $t$ do
        Calculate target location $L_t(u_i)$ which is the closest point to $L^*(u_i)$ without breaking the connection with $u_i$'s left and right neighbours $u_{i-1}$ and $u_{i+1}$.
        If $|L_t(u_i) - L^*(u_i)| > \delta$, then move to $L_t(u_i)$.
    end for
until all nodes have no further movement.
```

### *6.2 Protocol SAMCF*

The second protocol, SAMCF, selects a dummy sink as in SAMCC. Once a dummy sink is chosen, the intermediate nodes will move toward their optimal locations as much as possible without breaking the connections with their left and right neighbours. The details of this algorithm are shown in Algorithm SAMCF.

As said in its protocol, SAMCF will not disrupt the communication between neighbours when the intermediate nodes are moving to their optimal locations once a dummy node $d'$ is set for each source $s$.

### *6.3 Protocols SAMCM and SAMCD*

When O-SAS is applied to MCM or MCD, the real sink selects a dummy sink like the previous two protocols, and then the intermediate nodes will move according to MCM or MCD. SAMCM cannot guarantee the connectivity between communicating neighbours, and SAMCD has a slow convergence process. The oscillation problem in MCD is still solved by a threshold $\delta$. They are included here for comparison in the next section.

## 7 Simulation and performance analysis

### *7.1 Simulation settings*

We implement the sink-anonymity mobility control protocols (SAMCC, SAMCF, SAMCM and SAMCD) using a self-implemented simulator written in C language. In this paper, we focus on the mobility control model that is not affected by traffic patterns and throughput. Instead of using simulators such as NS2 and Omnet++, our simulator is sufficient for conducting experiments and obtaining results.

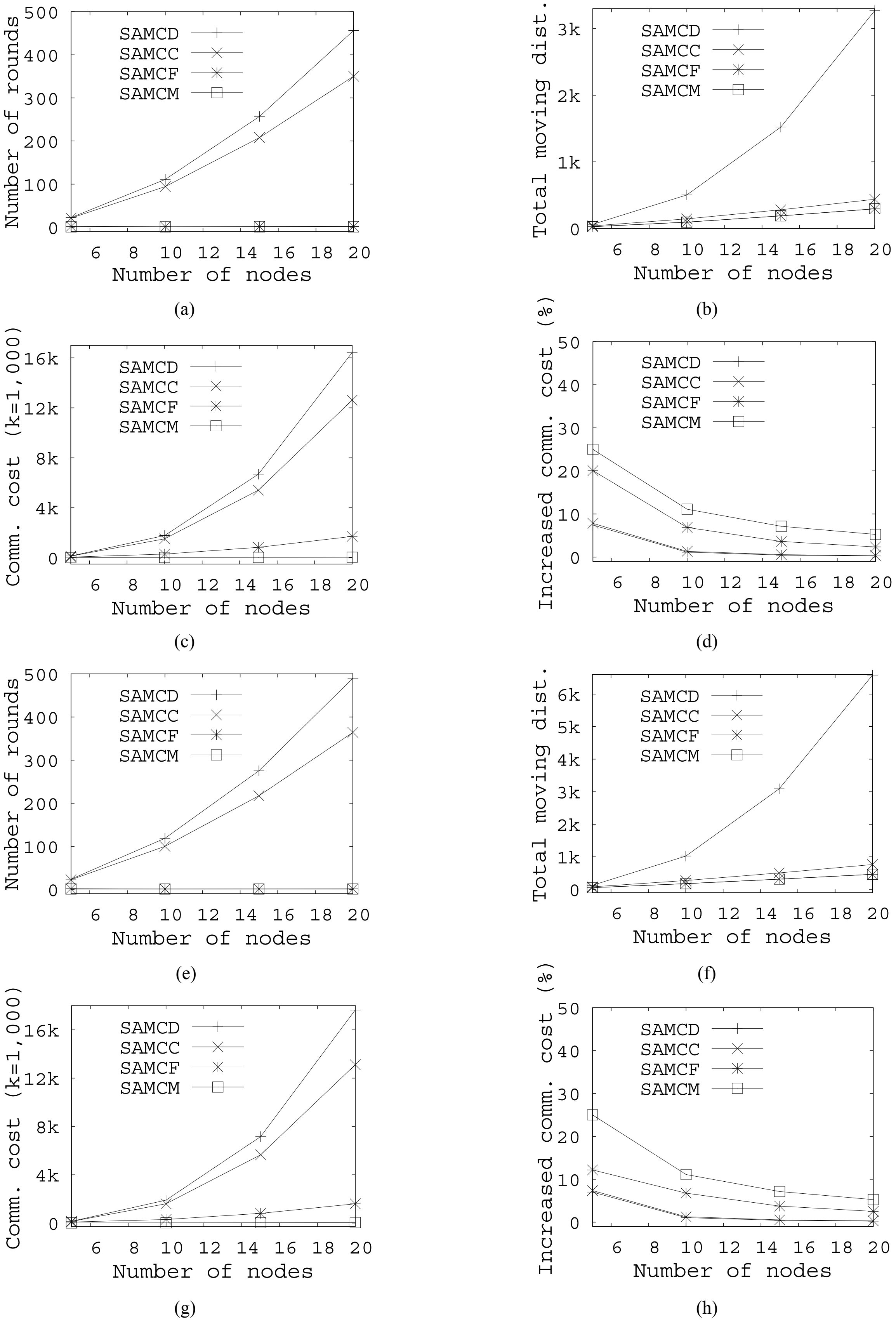
We conduct experiments using various network settings with different parameters. The initial locations of nodes are randomly generated in a 100 × 100 area. The number of nodes is set to 5, 10, 15 or 20, including the source and the destination. The communication range $R$ is 20 or 40 (Wu et al., 2006). The performance measurements are averaged over 10,000 experiments.

The performance of these protocols is measured by three metrics: convergence speed, energy consumption in node movement, and the communication cost. The convergence speed is obtained by the number of rounds of node movement needed to achieve stabilisation. The energy consumption in node movement is measured as the total moving distance of nodes. The communication cost of mobility control is calculated by the total number of messages exchanged between nodes.

#### *7.1.1 Simulation results on convergence*

Figure 11(a) and Figure 11(e) show the number of rounds of node movement for different algorithms when the communication range is set to 20 or 40 respectively with various node numbers. In the figures, SAMCD has the most rounds of node movement, SAMCC has less, SAMCF and SAMCM have the least. SAMCM has the fastest convergence because it allows nodes to move to their optimal locations in one round. In both figures, we can see that the curves of SAMCF almost overlap with those of SAMCM. This shows that SAMCF can converge surprisingly fast. It almost reaches the optimal result of SAMCM.

**Figure 11** Comparison of sink-anonymity mobility control protocols, (a) round # of node movement (R = 20) (b) total node moving dist. (R = 20) (c) comm. cost (R = 20) (d) increased comm. cost (%) (R = 20) (e) round # of node movement (R = 40) (f) total node moving dist. (R = 40) (g) comm. cost (R = 40) (h) increased comm. cost (%) (R = 40)



(a) (b) (c) (d) (e) (f) (g) (h)

### 7.1.2 Simulation results on total moving distance

Figure 11(b) and Figure 11(f) show the total moving distance of nodes during the convergence process using different algorithms when the communication range is 20 or 40 respectively with various node numbers. The results in these two figures match those of the number of rounds of node movement. One good result is that SAMCF is so close to SAMCM in terms of total moving distance that their curves are almost overlapped in the figures. As we know, SAMCM achieves the minimum total movement. Therefore, the total movement using SAMCF is extremely close to the minimum.

### 7.1.3 Simulation results on communication cost

Now, we look at the communication cost of these protocols. As shown in Figure 11(c) and Figure 11(g), the results of the communication cost match those of the number of rounds and total distance of node movement. SAMCD has the highest cost, SAMCC is the next, and SAMCF is very close to SAMCM which has the lowest cost.

In summary, these results show how good SAMCC and SAMCF are compared with SAMCM and SAMCD in convergence speed, energy consumption, and communication cost. This is especially true for SAMCF, which nearly reaches the best results of SAMCM.

## 7.2 Effects of embedding O-SAS in protocols

In this section, we show how the added security scheme O-SAS affects the convergence speed, total node moving distance and communication cost of the original protocols.

First, the convergence speed. When O-SAS is integrated into MCD, MCM, MCC and MCF, the resulting sink-anonymity mobility control protocols SAMCD, SAMCM, SAMCC and SAMCF will have the same convergence speed as the underlying protocols. This is because after a dummy sink is selected according to SAS, the intermediate nodes between the source and the sink will try to align themselves based on the position of the dummy sink. This process is no different in terms of number of rounds of node movement than using the real sink. So, adding security in these protocols does not affect the convergence property of them.

Second, the total moving distance. Similar to the convergence speed, built-in O-SAS has little impact on the total distance of node movement either.

Third, the communication cost. If O-SAS is integrated into MCD, MCM, MCF and MCC, the communication cost will increase as a result of extra message exchanges. We calculate the increased communication cost (in percentage) over each original protocol if security is embedded. From Figure 11(d) and Figure 11(h), the communication cost increases for all four protocols if security is used. The communication cost of SAMCM increases the most: e.g., 25% when the number of nodes is 5 and the communication range is 20; SAMCF is the next; SAMCC and SAMCD are the least. Since MCM and MCF are already low-cost protocols, anything added on to them will cause a greater increase in cost than those higher-cost protocols. As the number of nodes increases, the percentages fall sharply. Therefore, the built-in security will only bring trivial communication cost to the original protocols.

## 8 Conclusions

In this paper, two mobility control protocols with fast convergence, MCC and MCF, have been put forward to improve communication in WSNs. MCC speeds-up the convergence process by avoiding node's overreaction to the movement of its neighbours, whereas MCF reduces the convergence time by moving the nodes as close to their optimal positions as possible. Both protocols have embedded the information of the optimal locations of relay nodes into the mobility control. In addition, we have pointed out a unique privacy issue: the sink location privacy. To protect the sink node, we have proposed a new privacy-preserving scheme to free mobility control protocols from attacks that locate and sabotage the sink node. The privacy-preserving scheme obfuscates the sink location with dummy sink nodes. Analysis has shown that the scheme can effectively hide the sink location via anonymity. The scheme has also been integrated into the mobility control protocols without raising much additional overhead. The simulation and performance analysis have shown that the proposed privacy-preserving mobility control protocols, with the sink node well-protected, can reach near-optimal results in terms of convergence speed, energy consumption and the communication cost. All of these provide strong evidence of support in justifying the effectiveness of using mobility control to reduce energy-consumption to improve communication efficiency in WSNs. In the future, we will extend this work to enhance sink privacy with multiple path segments in mobility control. That is, there can be communications between one source and multiple destinations or between multiple sources and multiple destinations. The solutions involve resource sharing and competition. Another direction in the future is to apply our model to real applications by considering traffic and throughput in the network and verify the performance by NS2 or Omnet++.

## References

Capkun, S., Hamdi, M. and Hubaux, J. (2001) 'GPS-free positioning in mobile ad hoc networks', in *Proc. of the 34th Annual Hawaii International Conference on System Sciences*, pp.3481–3490.

Chaum, D.L. (1981) *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms*, Vol. 24, No. 2, pp.84–90.

Cortes, J., Martinez, S., Karatas, T. and Bullo, F. (2004) 'Coverge control for mobile sensing networks', *IEEE Transactions on Robotics and Automation*, Vol. 20, No. 2, pp.243–255.

Drissi, J. and Gu, Q. (2006) 'Localized broadcast authentication in large sensor networks', in *Proc. of International Conference on Networking and Services*.

Eschenauer, L. and Gligor, V.D. (2002) 'A key-management scheme for distributed sensor networks', in *Proc. of ACM CCS*, pp.41–47.

Estrin, D., Culler, D., Pister, K. and Sukhatme, G.S. (2002) 'Connecting the physical world with pervasive networks', in *Proc. of IEEE Pervasive Computing*, Vol. 1, pp.59–69.

Goldenberg, D., Lin, J., Morse, A., Rosen, B. and Yang, Y. (2004) 'Towards mobility as a network control primitive', in *Proc. of ACM MobiHoc*, pp.163–174.

Hu, D.L., Chan, B. and Bush, J.W.M. (2003) 'The hydrodynamics of water strider locomotion', *Nature*, Vol. 424, No. 7, pp.663–666.

Jadbabaie, A., Lin, J. and Morse, A. (2003) 'Coordination of groups of autonomous mobile agents using nearest neighbor rules', *IEEE Transactions on Automatic Control*, Vol. 48, No. 6, pp.988–1001.

Jiang, Z., Wu, J. and Kline, R. (2007) 'Mobility control for achieving optimal configuration in mobile networks', Technical report, Department of Computer Science, West Chester University.

Kahn, J.M., Katz, R.H. and Pister, K.S.J. (2000) 'Emerging challenges: mobile networking for 'smart dust', *Journal of Communications and Networks*, Vol. 2, No. 3, pp.186–196.

Kamat, P., Zhang, Y., Trappe, W. and Ozturk, C. (2005) 'Enhancing source-location privacy in sensor network routing', in *Proc. of IEEE ICDCS*, pp.599–608.

Karlof, C., Sastry, N. and Wagner, D. (2004) 'Tinysec: a link layer security architecture for wireless sensor networks', in *Proc. of International Conference on Embedded Networked Sensor Systems*, pp.162–175.

Li, L. and Halpern, J. (2001) 'Minimum-energy mobile wireless networks revisited', in *Proc. of IEEE ICC*, Vol. 1, pp.11–14.

Mao, Y. and Wu, M. (2005) 'Coordinated sensor deployment for improving secure communications and sensing coverage', in *Proc. of the 3rd Workshop on Security of Ad Hoc and Sensor Networks*, pp.117–128.

Mehta, K., Liu, D. and Wright, M. (2007) 'Location privacy in sensor networks against a global eavesdropper', in *Proc. of IEEE ICNP*.

Perkins, C. (2000) *Ad Hoc Networking*.

Poduri, S. and Sukhatme, G. (2004) 'Constrained coverage for mobile sensor networks', in *Proc. of IEEE International Conference on Robotics and Automation*, pp.165–171.

Rao, A., Papadimitriou, C., Shenker, S. and Stoica, I. (2003) 'Geographic routing without location information', in *Proc. of ACM MobiCom*, pp.96–108.

Reed, M., Syverson, P. and Goldschlag, D. (1998) 'Anonymous connections and onion routing', *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 4, pp.482–494.

Reif, J.H. and Wang, H. (1999) 'Social potential fields: a distributed behavioral control for autonomous robots', *Robotics and Autonomous Systems*, Vol. 27, pp.171–194.

Rodoplu, V. and Meng, T. (1999) 'Minimum energy mobile wireless networks', *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp.1333–1344.

Royer, E.M. and Toh, C.K. (1999) 'A review of current routing protocols for ad hoc mobile wireless networks', *IEEE Personal Communications*, pp.46–55.

Sibley, G.T., Rahimi, M.H. and Sukhatme, G.S. (2002) 'Robomote: a tiny mobile robot platform for largescale sensor networks', in *Proc. of IEEE International Conference on Robotics and Automation*.

Wang, W., Srinivasan, V. and Chua, K. (2005) 'Using mobile relays to prolong the lifetime of wireless sensor networks', in *Proc. of ACM MobiCom*, pp.270–283.

Wang, Y., Wu, H., Li, F. and Tzeng, N. (2007) 'Protocol design and optimization for delay/fault-tolerant mobile sensor networks', in *Proc. of IEEE ICDCS*, pp.7–15.

Wood, R. (2007) Available at http://www.eecs.harvard.edu/˜rjwood/media.html.

Wu, J., Cardei, M., Dai, F. and Yang, S. (2006) 'Extended dominating set and its applications in ad hoc networks using cooperative communication', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 17, No. 8, pp.851–864.

Xi, Y., Schwiebert, L. and Shi, W. (2006) 'Preserving source location privacy in monitoring-based wireless sensor networks', in *Proc. of IEEE IPDPS*.

Zhu, S., Xu, S., Setia, S. and Jajodia, S. (2003) 'Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach', in *Proc. of IEEE ICNP*, pp.326–335.