2009

Sink-Anonymity Mobility Control in Wireless Sensor Networks

Qijun Gu
Texas State University - San Marcos

Xiao Chen
Texas State University - San Marcos

Zhen Jiang
West Chester University of Pennsylvania, zjiang@wcupa.edu

Jie Wu
Florida Atlantic University

Recommended Citation
Gu, Q., Chen, X., Jiang, Z., & Wu, J. (2009). Sink-Anonymity Mobility Control in Wireless Sensor Networks. *WIMOB 2009, IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2009*, 36-41. http://dx.doi.org/10.1109/WiMob.2009.16

Sink-Anonymity Mobility Control in Wireless Sensor Networks

Qijun Gu, Xiao Chen
Dept. of Computer Science
Texas State University
San Marcos, TX 78666
{qg11, xc10}@txstate.edu

Zhen Jiang
Dept. of Computer Science
West Chester University
West Chester, PA 19383
zjiang@wcupa.edu

Jie Wu
Dept. of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
jie@cse.fau.edu

***Abstract*—As the technology of mobile sensors advances, mobility control becomes a viable option that can be utilized to minimize energy consumption in wireless sensor networks (WSNs). A mobility control protocol re-deploys mobile sensors to optimal positions to minimize energy consumption for communication. We identify a unique privacy issue in mobility control protocols that discloses the physical location of the sink node to intruders in WSNs. To protect the sink node, we propose a new privacy-preserving scheme to secure mobility control protocols against attacks that locate and sabotage the sink node. The privacy-preserving scheme obfuscates the sink location with dummy sink nodes. Analysis shows that the scheme can effectively hide the sink location via anonymity. The scheme can also be easily integrated into current mobility control protocols without raising much additional overhead. The performance simulation and analysis show that the mobility control protocols with sink-anonymity have the same near-optimal fast convergence process and close-to-minimum energy consumption as existing protocols but with the sink node well-protected.**

***Index Terms*—anonymity, mobility control, privacy-preserving, wireless sensor networks, sink location protection**

I. Introduction

As mobility becomes readily available to sensors [1], recent studies on using mobility as a control mechanism to minimize energy consumption [2]–[6] 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 showed that the saved energy in communication can compensate for the energy consumption in movement, thereby reducing the overall energy consumption of sensors. These protocols also ensure that the communication among sensors will not be disrupted when sensors are moving to their best locations.

In order for mobile sensors to find their most power-efficient locations, mobility control protocols yield the location information of the sink node to mobile sensors. Such information disclosure endangers the sink node, because attackers can easily obtain the location information via eavesdropping packets or capturing nearby sensors. The sink node in a sensor network is crucial for gathering, aggregating and transferring sensor information. If the sink node is located and destroyed, the network covered by the destroyed sink node will not function. Therefore, protecting the location of the sink node is one of the critical security issues in the effort to safeguard WSN operations. Nevertheless, the *protection of the sink location* can hardly be achieved using existing security mechanisms, such as packet encryption, key management, etc. At the same time, a scheme for sink protection should not affect normal sensing, communication and mobility control tasks that require knowledge of the sink location. To address this privacy issue, we propose a novel privacy-preserving scheme, named *sink-anonymity scheme*, that uses dummy sinks to deceive attackers and hide the sink location information in mobility control protocols. By hiding the true sink location, the cost of locating the sink node will be increased and baffle the attackers.

The contributions of the paper are threefold. (i) The privacy of the sink location is a unique issue in mobility control in WSNs. 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, etc. (ii) The privacy-preserving scheme is the first work to address the sink location privacy issue in mobility control. We show that the proposed privacy-preserving scheme has Φ-anonymity on the sink location. (iii) This scheme can be readily integrated into the mobility control protocols to enhance their security. The simulation shows that the mobility control protocols with sink-anonymity have the same near-optimal fast convergence process and close-to-minimum energy consumption as existing protocols.

The rest of the paper is organized as follows. Section II summarizes related privacy issues in WSNs and existing work on mobility control protocols. Section III provides the background information on mobility control and the privacy problem of the sink location. Section IV presents the privacy-preserving scheme and proves its Φ-anonymity on the sink location. Section V shows how to apply the privacy scheme in current mobility control protocols. Section VI shows the results of simulation and analysis on performance and the overhead of the privacy-enhanced mobility control protocols. Finally, Section VII concludes the paper.

II. Related Works

A. Privacy and K-anonymity

Privacy research was mainly conducted in the context of information privacy and anonymity. For example, packets and traffic patterns should never disclose identity information. A

few schemes [7]–[9] have been proposed on *source location privacy* in sensor networks. The main ideas of these schemes can be summarized 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 source nodes (real or fake) 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 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 behavior of the real object and thus hide the location of the real object. In this paper, we are interested in the problem of *sink location privacy* in mobility control, which is different from the source location privacy because the sink is usually the destination of routes in WSNs. New schemes are needed to ensure the privacy of the sink node in mobility control protocols.

The sink-anonymity problem in mobility control is related to the information anonymity in the data privacy research area. A formal privacy-preserving model named *K-anonymity* [10], [11] has been proposed that the record of an individual, upon being released to a query, is hidden in a group of at least k records with other individuals. Thereby, the privacy of the individual can be protected since the release of the record cannot be distinguished from at least $k-1$ individuals whose information also appears in the release. Various schemes [12], [13] have been proposed to efficiently create K-anonymity data sets. In this paper, we propose the Φ-anonymity model for sink location privacy. Unlike the K-anonymity model, the Φ-anonymity scheme does not create a fixed number of nodes to disguise the true sink node. Instead, the sink-anonymity scheme finds a continous area Φ such that the sink node could be hidden at any position inside Φ.

B. Mobility Control

Using mobility as a control primitive to minimize energy consumption in communication has been studied before. [2] proves 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 intermediate nodes must lie entirely on the line between the source and destination, and that the intermediate nodes must be evenly spaced along the line. The detail of the control algorithm is in Algorithm MCM (**M**obility **C**ontrol with **M**inimum total moving distance) [14]. MCM has a very nice property: the total moving distance of nodes in MCM is minimum.

As shown in MCM, intermediate nodes move to their optimal locations in one round. That can disconnect communicating neighbors [6]. To address this problem, a distributed algorithm is introduced in Algorithm MCD (**M**obility **C**ontrol **D**amped) [2]. In MCD, an intermediate node always only moves towards the average of its two neighbors, instead of reaching its optimal location in one round. It is proved that the connection between communicating neighbors using MCD will not be broken [2]. This algorithm suits the distributed environment because it only uses one-hop location information that is exchanged between a node and its left and right neighbors. However, MCD has a problem of slow convergence. That is, it takes nodes many rounds to reach their optimal locations. This is due to the fact that each node has to move as its two neighbors move no matter whether it is towards or away from its optimal location. To speed up the convergence process without loosing the connectivity between communicating neighbors, two quick convergence mobility control protocols are proposed [6]: MCC (***M**obility **C**ontrol quick **C**onvergence*) and MCF (***M**obility **C**ontrol **F**ast convergence*). Both protocols use the optimal location information of the intermediate nodes calculated by MCM before the convergence process starts. MCC speeds up the convergence process by avoiding the overreaction of a node to the movement of its neighbors, while MCF reduces the convergence time by moving the nodes as closer to their optimal positions as possible. In this paper, we will enhance MCC and MCF by the proposed privacy-preserving scheme to protect the sink location.

Algorithm MCM [14]: **M**obility **C**ontrol with **M**inimum total moving distance.

1: The source node s sends $L(s)$ and its label 0 to u_1. When each intermediate 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 intermediate node u_i, once both $L(s)$ and $L(d)$ are received, set $L^*(u_i) = L(s) + i \times \frac{L(d)-L(s)}{n}$ and move u_i to $L^*(u_i)$.

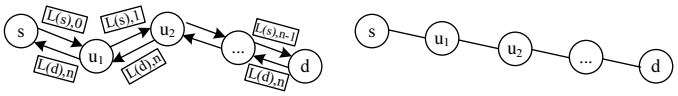
III. Background and Problems

In this section, we discuss the background of mobility control protocols and then introduce the privacy problem that motivates our work.

A. Mobility Control

To discuss mobility control, we assume that all sensor nodes have the same transmission range. Neighboring nodes can share their location information by exchanging short messages. Location information can be provided by GPS or other positioning algorithms such as the one in [15]. To simplify the discussion, we describe the protocols in a synchronous, round-based system. All the protocols presented in the paper can be extended to an asynchronous system. For security, we also assume some security schemes [16] are deployed in WSNs so that each mobile node can authenticate its own location information. When a node forwards another node's location information, the information cannot be modified and can be verified.

We assume that a path from the source s to the destination d (the sink node) has already been discovered using a routing protocol, e.g., a greedy routing protocol or one of the ad hoc



(a) before mobility control (b) after mobility control

Fig. 1. Mobility control

routing protocols. We also assume that neither s nor d are moving during mobility control. Otherwise, the path is always broken and a new routing path needs to be established. We label the nodes from the source to the destination as $0, 1, \cdots, n$. Node u_0 is the source s with location $L(s)$, node u_n is the destination d with location $L(d)$, and nodes $u_1, \cdots, u_{n-1}$ are intermediate nodes. For each node $u_i, 1 \leq i \leq n-1$, node u_{i-1} is its left neighbor and node u_{i+1} is its right neighbor.

With mobility control, the most power-efficient location $L^*(u_i)$ of u_i can be calculated [2] as

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

Once $L(s)$, $L(d)$, i and n are known, each node can calculate its optimal location according to Equation (1). Distributing $L(s)$, $L(d)$ and n to each node can be integrated into a routing protocol to reduce overhead. Figure 1 illustrates this process. When a sensor s is trying to establish a route to sink d, it sends a routing request to d. At the same time, it also sends its $L(s)$ and its label 0 along with the request. Each intermediate node will do the same thing until the message reaches d. Then, d sends a reply message with its $L(d)$ and the hop count of the path n back to s. When each intermediate node u_i has $L(s)$, $L(d)$, n and its own label, it can calculate its most power-efficient position according to Equation (1). All intermediate nodes can thus move to their best positions using a mobility control protocol such as MCM, MCD, MCC, or MCF and finally the path is formed as shown in Figure 1(b).

B. Privacy Issue in Mobility Control

The sink privacy issue with the existing protocols lies in the fact that (a) all sensors in a path need the sink location $L(d)$ to compute the optimal locations of the intermediate nodes and (b) a segment of a formed path can exposes the location of all nodes on the path. As the sink location is public to all the nodes along the path, the challenge is that traditional security mechanisms cannot help to protect 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. We believe and assume that attackers have the ability to capture and compromise any sensor node nearby and thus obtain any credential from the captured node.

Anonymity of sink location is different from anonymity of identity. 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 target to find the location of the sink node by analyzing packets transmitted near them and destroy it. Hence, the objective of our work is to hide the sink location.

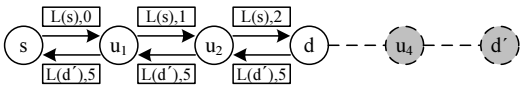


Fig. 2. Dummy sink node

We are aware that attackers can use other methods to locate the sink node as well. For instance, attackers can use special equipment to trace packets in a traffic flow hop by hop until reaching the sink node. However, we argue that such an attack method may not be feasible or may be too demanding in a complicated environment. For example, a sensor network may be deployed in a battle field. Tracing down a traffic flow will possibly expose and endanger the attackers themselves. Therefore, we believe that by hiding the sink location, the cost of such attacks may baffle the attackers.

IV. Sink-anonymity in Mobility Control

In this section, we present the privacy-preserving scheme to tackle the unique security issue known as *sink location privacy*. We first illustrate our idea that uses a dummy sink node in mobility control and show how the true sink node should choose the dummy sink node. Then, we propose Φ-anonymity to formalize the privacy problem and illustrate that the privacy problem can be handled by different strategies. Finally, we come up the sink-anonymity scheme that can fully obfuscate the sink location and prove its Φ-anonymity.

A. Dummy Sink Node

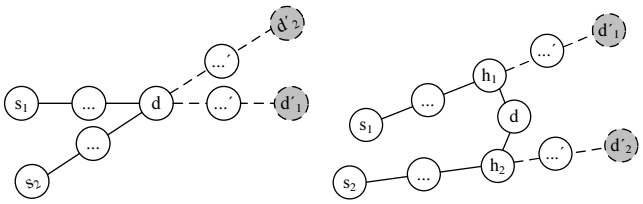
Our basic idea is to use a dummy node d' to hide the true sink location information $L(d)$ from all the nodes on the path. When sink node d receives the source location $L(s)$ and the total hop count n of the path from its previous node, it does not send back its real location $L(d)$. Instead, it creates a dummy node along the extension line of the path. The extended path ends at a dummy node $d' = u_m$ $(m > n)$ and includes $m - n - 1$ extra dummy intermediate nodes between d and d'.

For example, in Figure 2, the actual path includes four nodes in solid circles from source s to sink d. d creates two dummy nodes u_4 and d' and claims that the destination of the path is d'. Then, d sends back $L(d')$ as the sink location and $m = 5$ as the total hop count of the path from s to d'. $L(d')$ can be determined by Equation (2). Hence, no intermediate node on the path knows the true location $L(d)$ and the hop count n. Note that d can choose any $L(d')$ and m as long as Equation (2) holds.

$$L(d') = L(s) + \frac{m}{n}(L(d) - L(s)) \tag{2}$$

Upon receiving $[L(d'), m]$, all the nodes on the path compute their best locations similar to Equation (1), but substitute $L(d)$ with $L(d')$ and n with m. Hence, the best location $L^*(u_i)$ for node u_i is

$$L^*(u_i) = L(s) + i \times \frac{L(d') - L(s)}{m} \tag{3}$$



(a) Sink-On-Path Selection (b) Sink-Off-Path Selection

Fig. 3. Dummy node selection approaches

Theorem 1 proves that using a dummy node d' still guarantees that all the intermediate nodes will move to their most power-efficient locations as in Equation (1). In the example of Figure 2, suppose $L(s) = 0$ and $L(d) = 9$. When d receives $L(s)$ and label 2 from node u_2, it is supposed to send back $L(d)$ and $n = 3$. Using the dummy node d', d sends back $L(d') = 15$ and $m = 5$ instead of $L(d)$ and n. Now, the intermediate nodes u_1 and u_2 will use $L(d')$ and m to calculate their optimal locations using Equation (3). The results are 3 and 6 for u_1 and u_2 respectively. These optimal locations are the same as those using $L(d)$ and n.

Theorem 1: Using $L(d')$ and m, all the intermediate nodes have the same optimal locations as in Equation (1).

Proof: According to Equation (3), the optimal location of an intermediate node u_i is $L^*(u_i) = L(s) + i \times \frac{L(d')-L(s)}{m}$, when using $L(d')$ and m. Substitute $L(d')$ with the value in Equation (2), we get $L^*(u_i) = L(s) + i \times \frac{L(d)-L(s)}{n}$, which is the optimal location in Equation (1). ■

Note that $L(d)$ and n are two private values kept in the sink node d in the dummy sink scheme. If n is disclosed, $L(d)$ will be disclosed as well. For example, knowing n, if an attacker captures node u_i and obtains i, $L^*(u_i)$, $L(s)$, the attacker can derive the actual sink location $L(d)$ from Equation (1) as $L(d) = L(s) + \frac{n}{i}(L^*(u_i) - L(s))$.

Hence, a security requirement needs to be enforced in routing protocols so that a routing protocol in WSNs should not disclose the actual hop count of a path. We inspect several major routing protocols in WSNs [17], [18] and find that this security requirement can be satisfied because the destination node does not necessarily need to use the exact hop count of the true path. If the sink node chooses to include the dummy hop number m in routing packets, it will not affect the normal operations in routing and forwarding packets.

B. Dummy Sink Node Selection

Although using a dummy sink node can hide $L(d)$ as it seems, carefully selecting a dummy node is critical to ensure the sink location privacy. In Figure 3, we illustrate two dummy node selection approaches: *sink-on-path* and *sink-off-path*.

In the sink-on-path selection approach, the true sink node always makes itself on the path from the source to the dummy node. Hence, the sink location $L(d)$ is also the optimal location for the true sink node in the path, i.e. $L(d)$ is $L^*(d)$ that satisfies Equation (3). However, attackers can infer the actual sink location by compromising two nodes in two disjoint paths. This attack is illustrated in Figure 3(a). Assume that the sink is accepting information from two sources s_1 and s_2 via two disjoint paths. The sink claims two dummy sink locations $L(d'_1)$ and $L(d'_2)$ and dummy hop counts m_1 and m_2 for the two paths. If an attacker can compromise two nodes u_i and u_j on these two paths respectively, he can obtain $L(d'_1)$, $L(d'_2)$, m_1, m_2, $L(s_1)$ and $L(s_2)$. The attacker can then find the intersection of the two paths, which is the location of the true sink node d.

To counteract this attack caused by the sink-on-path selection, we propose the sink-off-path dummy node selection approach. The sink node d picks a one-hop neighboring node h and a dummy node d' such that h satisfies Inequality (4) and d' satisfies Equation (5), 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 (4) 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 (5) is a transformation of Equation (3) and states that d', h and s are on the same path.

$$|L^*(h) - L(d)| \leq R \tag{4}$$

$$\frac{L(d') - L(s)}{n_{d'}} = \frac{L^*(h) - L(s)}{n_h} \tag{5}$$

The sink-off-path selection approach is illustrated in Figure 3(b). For the two disjoint paths, d selects h_1 and d'_1 for s_1 and h_2 and d'_2 for s_2. d is not on either path and is one-hop away from h_1 and h_2. However, d claims that it is the next hop to h_1 and h_2 on their paths. Hence, when h_1 (or h_2) receives a packet from s_1 (or s_2), it will forward the packet to its next hop which is d. That is, d can accept information delivered in both paths.

The sink-off-path dummy node selection approach has two privacy properties as proved by Theorems 2 and 3. We will use the sink-off-path dummy node selection approach as the baseline to develop our full privacy-preserving scheme.

Theorem 2: The sink-off-path dummy node selection approach does not disclose the true 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. ■

Theorem 3: The sink-off-path dummy node selection approach does not disclose the sink location, if the intersection point 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 intersecton point x does not disclose the sink location. ■

C. Φ-anonymity

Although attackers cannot directly obtain the true sink location by compromising sensors along multiple disjoint paths,

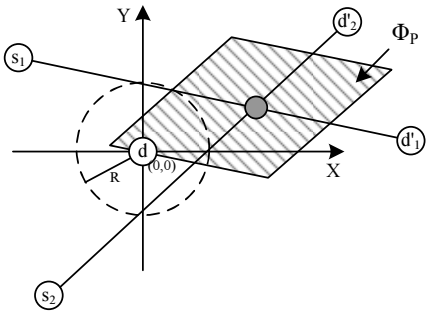


Fig. 4. Illustration of Φ_P-anonymity

they may try to find a *proximity area* surrounding the true sink node, because the sink node is surely within one-hop distance to the paths. Hence, we propose a formal privacy model to analyze the privacy achieved by the sink-off-path dummy node selection approach. The model can help the sink node to compare the privacy when different dummy nodes are selected. The model also leads to the sink-anonymity mobility control protocols presented in Section V.

As shown in Figure 4, we put the sink node in the origin. Assume that attackers have found N disjoint paths. Each path equation i is denoted as $y = k_i x + c_i, \ \ for \ \ 1 \leq i \leq N$.

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 $\frac{|c_i|}{\sqrt{1+k_i^2}} \leq R$.

Similarly, in order to determine whether the sink node is at the location (X, Y), 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 (6), 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).

$$\frac{|c_i + k_i X - Y|}{\sqrt{1 + k_i^2}} \leq R \ \ for \ \ 1 \leq i \leq N \tag{6}$$

In the example of Figure 4, the proximity area is the shadowed area in which any position (X, Y) satisfies Inequality (6), because it is within the communication range to either of the two disjoint paths. Hence, the proximity area is the achieved privacy against the two compromised paths.

Given such a proximity area, we define Φ-anonymity as below. Accordingly, the proximity area in Figure 4 is a Φ_P-anonymity, where P is the set of the two disjoint paths.

Definition 1: Let Φ be a proximity area and P be the set of all disjoint paths known to attackers. Φ is said to satisfy Φ_P-anonymity if and only if Φ is the maximum proximity area in which any location (X, Y) satisfies Inequality (6) for all paths in P.

Definition 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 (6) for all paths in P^*.

The relation of Φ_P-anonymity and Φ-anonymity is shown by Theorem 4 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 have on the true sink location. However, the minimum proximity area that attackers can achieve is the Φ-anonymity area.

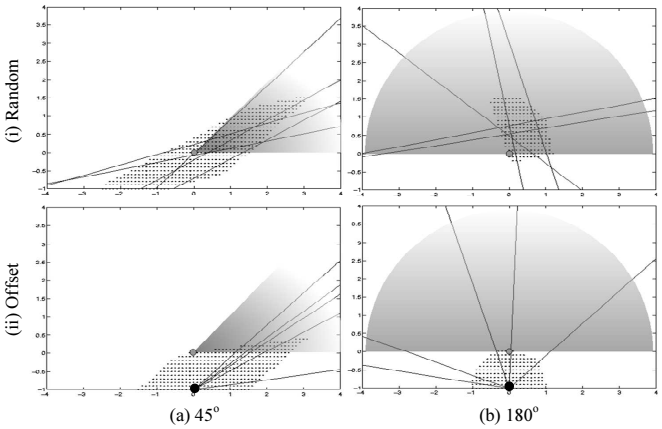


Fig. 5. Sink coverage and path selection approaches

Theorem 4: A Φ-anonymity area is the minimum in all Φ_P-anonymity areas, i.e. $\forall \Phi_P$, $\Phi \subseteq \Phi_P$.

Proof: Assume we can find a P and a Φ_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^*. But, a path $p \in P$ exists that x is further 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 to $P \nsubseteq P^*$.

Therefore, the theorem is proved by contradition. ■

D. Sink-anonymity Schemes

The Φ-anonymity area is critical to the privacy of sink location. The larger the Φ-anonymity area is, the better the true sink node is protected. According to Definition 2, the shape and the size of the Φ-anonymity is determined by P^*. In other words, the disjoint paths selected by the sink node determine the privacy of the sink location.

We propose two sink-anonymity schemes (shown in R-SAS and O-SAS) that use different disjoint path selection approaches and result in different privacy protection. 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 θ. Figure 5 illustrates the fans of $\theta = 45^o$ and $\theta = 180^o$ (the gray areas). When $\theta = 180^o$, paths to the sink node may come from all directions. Thereby, paths in a fan of $\theta > 180^o$ are the same as paths in a fan of $\theta = 180^o$. Note that the fan area of a sink node is normally determined by network deployment or task assignment. We assume the sink node has θ as a parameter in mobility control.

Both of the sink-anonymity schemes (SASs) use the sink-off-path dummy node selection approach. The difference is that R-SAS uses the *random disjoint path selection* approach while O-SAS uses the *offset disjoint path selection* approach.

R-SAS: Random **S**ink-**A**nonymity **S**cheme

1: **for** Each requesting sensor s **do**
2: The sink node uses the sink-off-path approach to select a one-hop neighboring node h and a random dummy node d' such that
(a) h satisfies Inequality (4),
(b) d' satisfies Equation (5).
3: **end for**

O-SAS: Offset **S**ink-**A**nonymity **S**cheme

1: The sink picks an offset point x in the offset area and keeps the offset from x to d as a secret.
2: **for** Each requesting sensor s **do**
3: The sink node uses the sink-off-path approach to select a one-hop neighboring node h and a random dummy node d' such that
(a) h satisfies Inequality (4) and Equation (7),
(b) d' satisfies Equation (5).
4: **end for**

The random selection approach makes each selected path go through a randomly positioned one-hop neighbor and point to a randomly selected dummy node. The offset selection approach 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 5). Hence, in addition to Equation (5) and Inequality (4), all the paths selected by the offset approaches also satisfy Equation (7) which states that the offset point x is on the paths. Note that x is neither a dummy node nor a true node.

$$\frac{Y_x - Y_s}{X_x - X_s} = \frac{Y_h - Y_s}{X_h - X_s} \tag{7}$$

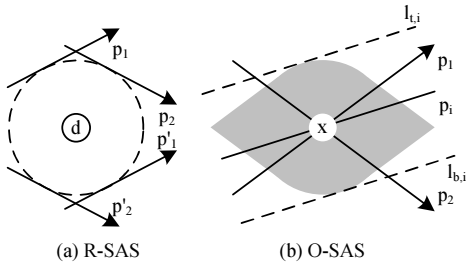
Figure 5 shows examples of the Φ_P-anonymity area (the dotted areas) when attackers know a set P of five disjoint paths. The dotted areas illustrate several privacy properties. First, the actual sink node could be at any location in the dotted areas. Knowing the dotted area does not necessarily disclose 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.

E. Privacy Analysis of R-SAS and O-SAS

To analyze the privacy achieved by R-SAS and O-SAS, we need to identify the Φ-anonymity areas in the proposed schemes. Theorem 5 shows that R-SAS does not provide any privacy protection to the true sink node if attackers comprise sensors in sufficient disjoint paths. On the contrary, Theorem 6 shows that O-SAS can achieve Φ-anonymity to protect the sink location.

Theorem 5: The Φ-anonymity area of R-SAS could be as small as the true sink node.

Proof: Because the sink node randomly selects dummy nodes, it possibly selects two pairs of parallel paths $\{p_1, p_1'\}$ and $\{p_2, p_2'\}$ as shown in Figure 6(a). Thereby, 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, Φ_P includes only the sink node. Because $\Phi \subseteq \Phi_P$ as in Theorem 4, Φ includes only the sink node. ■



(a) R-SAS (b) O-SAS

Fig. 6. Analysis of Φ-anonymity

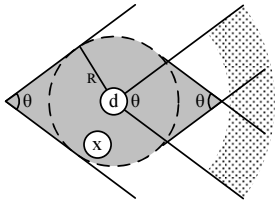


Fig. 7. Offset Area

Theorem 6: Let p_1 and p_2 be the two outmost paths in O-SAS as shown in Figure 6(b) such that all paths in P^* are within the area bounded by the two paths. The Φ-anonymity area of O-SAS is the gray area in Figure 6(b).

Proof: For any $p_i \in P^*$, let $P_i = \{p_i\}$. Find two parrallel lines $l_{t,i}$ and $l_{b,i}$ as in Figure 6(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 Φ_{P_i}-anonymity area.

We rotate p_i from p_1 to p_2. For each instance of p_i, we find the corresponding Φ_{P_i}-anonymity area. The overlaping area of all the Φ_{P_i}-anonymity areas, which is the gray area in Figure 6(b), is the Φ-anonymity area. ■

For O-SAS, the solid gray area in Figure 7 shows where the sink node can select an offset point x. The farthest distance between the offset point and the sink node is $\frac{R}{sin(\theta/2)}$. If $\theta \leq 60^o$, the offset point could be more than two hops away from the true sink node, while any path passing the offset point is one-hop away from the true sink node.

F. Simulation and Comparison of R-SAS and O-SAS

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

pe basically tells how far away the center 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. Thereby, from a defender's perspective, larger pe, pm, and pa indicate better privacy.

Figure 8 summarizes the measurement of privacy in three metrics. We study the situations where θ is 45^o, 90^o or 180^o. We measure the privacy, assuming that the sink node has a few disjoint paths ranging from 3 to 19 in its covered area.

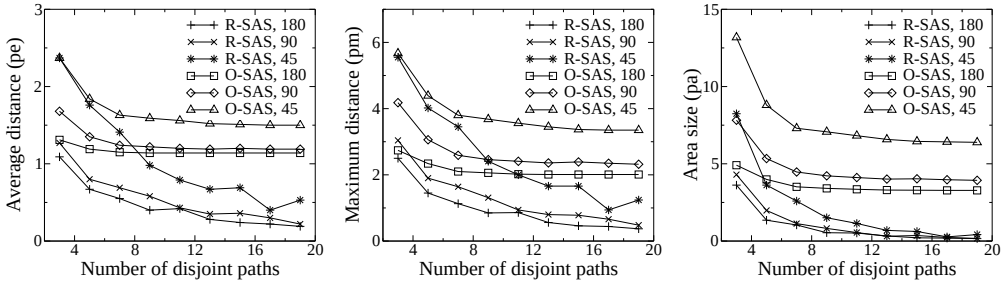


Fig. 8. Comparison of Φ-anonymity of R-SAS and O-SAS

The results are normalized as the communication range of the sink node is set to 1. All data points are averaged over 30 experiments.

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 3 disjoint paths. Thereby, attackers can estimate a very close location to the sink node 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 trying more disjoint paths with the application of O-SAS.

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

V. 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 well protect the true sink node and ensure the connectivity between communicating neighbors. We also apply the O-SAS scheme to MCM and MCD. They will be used in our simulation for comparison.

A. 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 by MCM, and if the distance between its new position (which is calculated as the average of its two neighbors' 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.

Algorithm SAMCC: Sink-Anonymity MCC.

1: The sink node d picks a dummy sink d' for each source s according to O-SAS.
2: The sink sends the location $L(d')$ back to the source via its neighbor h.
3: Apply MCM to obtain the optimal location $OL(u_i)$ for each intermediate node u_i.
4: **repeat**
5: **for** Each intermediate node u_i **do**
6: Exchange $L(u_i)$ with u_{i-1} and u_{i+1}.
7: Receive $L(u_{i-1})$ and $L(u_{i+1})$. Set $L^*(u_i) = \frac{L(u_{i-1})+L(u_{i+1})}{2}$.
8: If $|L^*(u_i) - OL(u_i)| > |L(u_i) - OL(u_i)|$ no movement.
9: Else if $|L^*(u_i) - L(u_i)| \geq MDPR$, move to $L^*(u_i)$.
10: **end for**
11: **until** All nodes stop

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 according to MCM. During the process, SAMCC will not disconnect communicating neighbors [6].

B. 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 neighbors. In this way, for each node, there is no extra movement. The details of this algorithm are shown in Algorithm SAMCF.

Same as SAMCC, SAMCF will not disconnect communicating neighbors when real intermediate nodes are moving to their optimal locations once the dummy node d' is set for a source s.

C. Protocols SAMCM and SAMCD

When O-SAS is applied to MCM or MCD, the real sink selects a dummy sink as the previous two protocols and then the intermediate nodes will move according to MCM or MCD. SAMCM cannot guarantee the connectivity between communicating neighbors, and SAMCD has a slow convergence process. They are included here for the comparison in the next section.

Algorithm SAMCF: Sink-Anonymity MCF.

1: The sink node d picks a dummy sink d' for each source s according to O-SAS.
2: The sink sends the location $L(d')$ back to the source via its neighbor h.
3: Apply MCM to obtain the optimal location $OL(u_i)$ for each intermediate node u_i.
4: **repeat**
5: **for** Each intermediate node u_i **do**
6: Calculate target location $L^*(u_i)$ which is the closest point to $OL(u_i)$ without breaking the connection with u_i's left and right neighbors u_{i-1} and u_{i+1}.
7: If $|L^*(u_i) - L(u_i)| > MDPR$, move to $L^*(u_i)$.
8: **end for**
9: **until** All nodes stop

VI. Analysis and Simulation

In this section, we conduct simulations to measure the performance of the SAMCD, SAMCM, SAMCC, and SAMCF protocols. We show how the added sink-anonymity scheme affects the convergence speed, total nodes movement, and communication cost of the original ones.

1) Simulation Settings: In our simulation, we use three metrics: the convergence speed, the energy cost, and the communication cost of the stabilization process. In a synchronous, round-based system, the speed of achieving stabilization is measured by the number of rounds of node movement needed for convergence. The energy cost of mobility control protocols primarily comes from the energy consumed in node movement which is determined by the distance a node moves. In our experiments, the total distance of movement of all the nodes is used as a metric for the energy cost of mobility control protocols. The communication cost of mobility control protocols is calculated by the total number of messages exchanged among nodes in this paper.

For each algorithm, the number of rounds, the total distance of node movement, and the total messages exchanged are calculated. In our experiments, we try various network settings with different parameters. The number of nodes tried is 5, 10, 15 and 20, including the source and the destination. The transmission range used is 20 and 40 [19]. The initial locations of the nodes are randomly generated.

2) Simulation Results on Convergence: Figures 9(a) and 9(e) show the number of rounds of node movement for different algorithms when the transmission range is set to 20 and 40 respectively with the number of nodes varied. 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. From either figure, we can see that the line of SAMCF is almost overlapped with that of SAMCM. This shows that SAMCF can converge surprisingly fast. It almost reaches the optimal result of SAMCM.

3) Simulation Results on Energy Cost: Figures 9(b) and 9(f) show the total distance of node movement during the convergence process using different algorithms when the transmission range is 20 and 40 respectively with the number of nodes varied. The results in these two figures match those of the number of rounds of node movement. One very good result is that SAMCF is so close to SAMCM in terms of the total distance that their lines overlap in the figures. As we know, SAMCM achieves the minimum total movement. Therefore, the total movement using SAMCF is extremely close to the minimum.

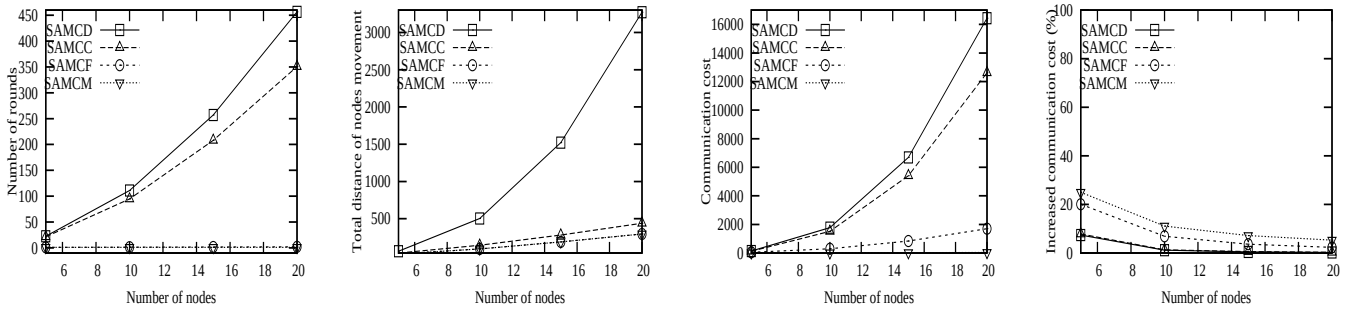
4) Simulation Results on Communication Cost: Now we look at the communication cost of these protocols. As shown in Figures 9(c) and 9(g), the results of the communication cost match those of the number of rounds and total distance of node movement. SAMCD has the highest cost, SAMCC is the next,and SAMCF is very close to SAMCM which has the least cost.

In summary, these results show us how good SAMCC and SAMCF are compared with SAMCM and SAMCD in terms of convergence speed and energy consumption. Especially SAMCF, it almost reaches the best results by SAMCM.

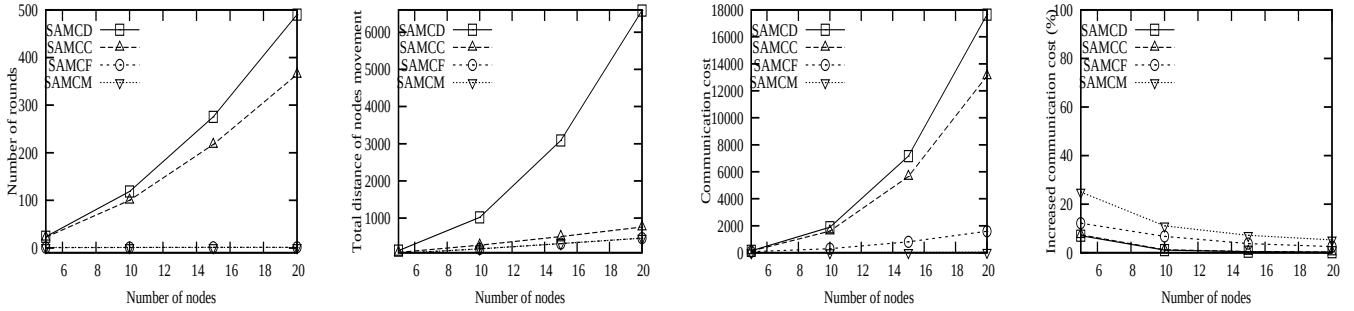
5) Effects of Embedding O-SAS in Protocols: In this section, we show how the added O-SAS affects the convergence speed, total nodes movement, and communication cost of the original protocols.

When O-SAS is integrated into the MCD, MCM, MCC, and MCF protocols, the resulting privacy-preserving mobility control protocols SAMCD, SAMCM, SAMCC and SAMCF will have the same convergence speed as the protocols they are built on. This is because after a virtual sink is created according to O-SAS, the intermediate nodes between the source and the sink will try to align themselves based on the position of the virtual sink. This process is no different in terms of number of rounds of node movement than using the real sink. Therefore, adding the security in these protocols does not affect the convergence property of them. Similarly, built-in O-SAS will not affect the total distance of node movement either.

Next we show how the embedded sink-anonymity scheme affects the communication costs of the protocols. If O-SAS is integrated into MCD, MCM, MCF, and MCC, the communication costs will increase because of the extra message exchanges. Here we calculate the increased communication cost (in percentage) over each original protocol if security is embedded. From Figures 9(d) and 9(h), we can see that the communication costs have increased for all three protocols if security is used. The communication cost of SAMCM increases the most: for example, 25% when the number of nodes is 5 and the transmission range is 20; SAMCF is the next; SAMCC and SAMCD are the least. This is because MCM and MCF are already low-cost protocols, anything added on will be more outstanding in increased costs 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 costs to the original protocols.



(a) Round # of node movement (r=20) (b) Total node movement dist. (r=20) (c) Comm. cost (r=20) (d) Increased comm. cost (%) (r=20)



(e) Round # of node movement (r=40) (f) Total node movement dist. (r=40) (g) Comm. cost (r=40) (h) Increased comm. cost (%) (r=40)

Fig. 9. Comparison of mobility control protocols

VII. Conclusion

In this paper, we have identified a unique privacy issue in mobility control that discloses the physical location of the sink node to intruders in WSNs. To protect the sink node, we have proposed a new privacy-preserving scheme to secure mobility control protocols against attacks that locate and sabotage the sink node. The privacy-preserving scheme can obfuscate the sink location with dummy sink nodes. The analysis has shown that the scheme can effectively hide the sink location with Φ-anonymity. The scheme has also been integrated into current mobility control protocols without raising much additional overhead. The performance simulation and analysis have shown that the mobility control protocols with sink-anonymity have the same near-optimal fast convergence process and close-to-minimum energy consumption as the existing ones but with the sink node well protected. In the future, we will extend this work to enhance sink privacy with multiple path segments in mobility control.

References

[1] V. Rodoplu and T. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1333–1344, 1999.

[2] D. Goldenberg, J. Lin, A. Morse, B. Rosen, and Y. Y, "Towards mobility as a network control primitive," in *Proc. of ACM MobiHoc*, 2004, pp. 163–174.

[3] L. Li and J. Halpern, "Minimum-energy mobile wireless networks revisited," in *Proc. of IEEE ICC*, vol. 1, 2001, pp. 11–14.

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

[5] Y. Wang, H. Wu, F. Li, and N. Tzeng, "Protocol design and optimization for delay/fault-tolerant mobile sensor," in *Proc. of IEEE ICDCS*, 2007, pp. 7–7.

[6] X. Chen, Z. Jiang, and J. Wu, "Quick convergence mobility control schemes in wireless sensor networks," in *Proc. of 10th Workshop on Advances in Parallel and Distributed Computational Models*, 2008.

[7] Y. Xi, L. Schwiebert, and W. Shi, "Preserving source location privacy in monitoring-based wireless sensor networks," in *Proc. of Parallel and Distributed Processing Symposium*, 2006.

[8] K. Mehta, D. Liu, and M. Wright, "Location privacy in sensor networks against a global eavesdropper," in *Proc. of IEEE ICNP*, 2007.

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

[10] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[11] A. Meyerson and R. Williams, "On the complexity of optimal K-anonymity," in *Proc. of ACM PODS*, 2004, pp. 223–228.

[12] R. Bayardo and R. Agrawal, "Data privacy through Optimal K-anonymization," in *Proc. of IEEE ICDE*, 2005, pp. 217–228.

[13] H. Park and K. Shim, "Approximate algorithms for K-anonymity," in *Proc. of ACM SIGMOD*, 2007, pp. 67–78.

[14] Z. Jiang, J. Wu, and R. Kline, "Mobility control for achieving optimal configuration in mobile networks," 2007.

[15] S. Capkun, M. Hamdi, and H. J., "Gps-free positioning in mobile ad hoc networks," in *Proc. of the 34th Annual Hawaii International Conference on System Sciences*, 2001, pp. 3481–3490.

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

[17] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proc. of ACM MobiCom*, 1999, pp. 174–185.

[18] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, 2003.

[19] J. Wu, M. Cardei, F. Dai, and S. Yang, "Extended dominating set and its applications in ad hoc networks using cooperative communication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 8, 2006.