A New Scheme for Source-location Privacy in Wireless Sensor Networks

Shaoquan Jiang, Min Li, and Zailiang Tang
(Corresponding author: Min Li)

Institute of Information Security, Mianyang Normal University
166 Mianxing Rd. West, High-Tech District, Mianyang 621000, China
(Email: shaoquan.jiang@gmail.com)
(Received May 16, 2017; accepted Aug. 26, 2017)

Abstract

In this work, we study the source location privacy in wireless sensor networks (SLP-WSN), where a source wishes to send a message to a base station while preserving the privacy of its location. This problem has a lot of applications including habitat monitoring and military surveillance. A well-known strategy is to divide the message transmission into two stages. The first stage routes the source message to a random position in the network while the second stage routes the source message to the base station through the shortest path. To preserve the source location privacy, the system must securely mask the trace of the routing. Thus, the first stage routing is essential. The literature proposed many approaches. However, they either assume a sensor's awareness of its location or have high energy consumption or have an obvious privacy weakness. In this work, we propose a new method for SLP-WSN without these defects. We first help a sensor node to obtain distances to three reference nodes by flooding. Then, we design the first stage routing probabilistically, where the next sensor is chosen according to a carefully designed distribution such that a sensor close to the random point has a higher probability. Our experiment shows that the energy consumption and the message delay only have a small factor expansion, relative to directly routing via the shortest path to the base station. It is also shown to have a good privacy.

Keywords: Information Security; Location Privacy; Security Protocol; Sensor Network Security

1 Introduction

With the advances of the network technology, wireless sensor networks (WSNs) are widely believed to be a good solution for monitoring unattended or dangerous environments. The network is formed with many small and cheap sensors [24]. Sample applications include environmental monitoring and military surveillance.

Due to its unattended nature, it is vulnerable to various attacks. The standard security problems such as confidentiality and authentication can be solved using well-known cryptographic techniques. However, some issues can not be solved cryptographically [10, 18, 23]. The location privacy is one of them. It can be well interpreted by the panda monitoring problem, where sensors are deployed in the panda's habitat to monitor its activity and report to the base station. However, since the transmission is wireless in nature, it can be eavesdropped by hunters who attempts to trace back to the panda through the message path. This tracing procedure does not need to know the meaning of a message (if encrypted) and hence it is not concerned with the secret keys. Thus, a careful design (especially the routing protocol) is needed to combat such an attack. In this sense, a regular routing scheme for WSN is certainly not enough.

1.1 Related Works

The representative technique for location privacy in WSN is phantom routing proposed by Ozturk *et al.* [21] (enhanced by Kamat *et al.* [8]), where the message routing consists of two stages: in stage one, it follows a directed random walk of h steps and in stage two, the message is routed to the base station via flooding or a single path. Li *et al.* [12, 15] extended the phantom routing by first sending the message to a random intermediate rode and then routing it to a network mixing ring (NMR) before sending it to the base station. Li *et al.* [13, 14] considered multi random intermediate nodes by choosing their polar coordinates as $(d_1, \alpha), (d_2, 2\alpha), \cdots, (d_n, n\alpha)$, where α is a random angle. A single path routing to a random intermediate point from a restricted area was studied in [16]. Yao and Wen [27] considered the random shortest path from the source to the base station. This approach does not have a good location-privacy as with high probability the physical routing path will stay around the line between the source and the base station and hence the attacker will be relatively easy to capture many packets and trace

back to the source in a hop-by-hop manner. The phantom routing technique was further studied in [4, 5, 11].

Location privacy for a receiver was studied in the literature. Although this is different from our problem, it can be regarded as the dual problem to the source location privacy. In [2, 6, 9], the protocol is based on a multi-path routing and a fake packet injection. However, their fake packet injection has a constant rate and hence is energy consuming. Jian *et al.* [7] considered the location privacy of a moving receiver in a wireless sensor network. Their method involves a fake packet injection by each intermediate sensor and the fake packet needs to transmit sufficiently far. Hence, their method is still energy-consuming. Luo *et al.* [17] proposed a variant of [7]. But they still need to forward the fake packet and hence is energy consuming again. Phantom routing and fake source techniques are combined in [3].

Yao *et al.* [28] proposed a new method for source-location privacy in WSN, where they considered the notion of *ring* that is the set of nodes with the same distance to the base station. They first routes the message to a random ring c and then routes it on the ring toward a fixed direction for some steps. Next, they routes to a random ring b to do the similar thing and finally routes the message directly to the base station. It is assumed that a node on a ring knows which neighbors are in a given direction and which are not. This can not be implemented only by a landmark (as done by the east-west separation in [8]) because nodes are placed circularly. So it requires the awareness of its own and neighbors' positions.

Yang *et al.* [26] considered the clustered wireless sensor network with a cluster head more powerful than a common sensor and the location-privacy is achieved through the faking message simulation by cluster heads. This method is not suitable for a homogenous network such as our setting since the message simulation will consume the power quickly. Location-privacy in wireless sensor network against a global eavesdropper was studied in [1, 19, 20, 22, 25, 29], where the adversary can eavesdrop the whole network and the privacy is achieved by fake packets. Since a global attacker is more powerful than our local eavesdropper, this method is relevant to our setting. However, it always results in a large energy consumption and we also feel that a global eavesdropper model is too strong. Hence, we will not consider such an adversary.

1.2 Contribution

In this work, we propose a new method for SLP-WSN. Our scheme lies in the well-known two-stage strategy: the first stage routes the message to a random intermediate position and the second stage routes the message to the base station. However, we do not assume any location information for each sensor (beyond the knowledge of three reference nodes' coordinates). Instead, we help a sensor to obtain distances (termed *hop distance* which is the number of hops between two nodes) to reference nodes and use the sensor's hop distance tuple as its location information. We also find out a method to estimate the hop distance between two positions while each position is represented by its hop distance tuple. Our scheme starts the first stage routing using a probabilistic method. Specifically, starting from the source, the next sensor is chosen probabilistically among the current sensor's neighbors, where the probability distribution is carefully designed such that a sensor close to the random intermediate position has a better chance of being selected. Our choice of the probability distribution allows the next-hop sensor is chosen probabilistically so that it is hard to trace back, while the message can still steadily move toward the random intermediate position. The second stage is a shortest path routing. Under our design, we evaluate its privacy and efficiency. We consider the message delay measure (called *PathRatio*), defined as the ratio of our routing path length to the shortest path from the source to the base station. Our *PathRatio* only has a small factor. We also consider the energy consumption ratio (termed *EnergyRatio*), defined as the length of total messages sent in the whole network to the length of total messages sent by nodes when only using the shortest path routing from the source to the base statio. Our scheme, *EnergyRatio* is equal to *PathRatio* and is small. We also consider the *SafetyPeriod*, defined as the number of source messages the source can send before its location privacy is broken. We build a model to quantify this and find our protocol has a good privacy. Finally, assuming the network is almost fully connected, our scheme can deliver the source message reliably. In comparison with existing works [8, 12, 13, 14, 15, 16, 21], our protocol either has a much smaller *EnergyRatio* or a better privacy or removes a node's awareness of its location; see Section 4.3 for details.

2 Model

We now formalize the source-location privacy in wireless sensor networks (SLP-WSN). This consists of the system model, assumptions, location privacy and efficiency measures.

2.1 System Model

SLP-WSN is a system that enables a source S to transmit a message m to a base station under the help of some sensors such that no adversary, who eavesdrops the traffic at some points, can determine the location of S. In our model, S is an entity that can communicate, where the motivation example is the soldier in a battle field or a sensor carried by a monitored animal. In some works, S is a sensor that monitors the environment such as a wild animal. Obviously, these two presentations have no essential differences in the location privacy technology. We use $v_1, \cdots, v_{n-1}$ to denote *sensors* and use v_0 to denote *base station*. For simplicity, a sensor or base station is

called a *node*.

We suggest a three-stage model for a SLP-WSN system, which consists of a deployment stage, a preprocessing stage and a message transmission stage. Details are as follows.

Deployment. In this stage, a system manager will deploy nodes in a desired area. The position of S is undetermined and it can move arbitrarily. For simplicity, we assume that the deployed area is planar with radius R. However, our work can be easily generalized to the three-dimensional case.

Preprocessing. In this stage, $v_0, \cdots, v_{n-1}$ jointly execute a protocol. At the end, each v_i obtains an internal state, which will be crucial for the next-stage protocol.

Message transmission. This is the main part of the system. It helps source S transmit a source message m to base station v_0. Toward this, S will find a node v_{i_0} nearby and send m to it. Then, v_{i_0} will find an adjacent node v_{i_1} and send m to it. This process continues until m reaches v_0.

2.2 Assumptions

Our system will make the following assumptions.

- We assume that the location of a sensor is fixed throughout the system. This assumption will be used to keep the internal state of a node obtained in the preprocessing stage unchanged. However, as long as the network topology does not change frequently, it can be relaxed by executing the preprocessing stage periodically. This will be further discussed in Section 5.2. This slow changing topology is suitable for applications such as the habitat monitoring.
- The sensor network as an undirected graph is almost fully connected, where an edge (v_i, v_j) means that v_i is in the hearing range of v_j and vice versa. This in fact is the necessary assumption for any useful sensor network. We use the term "almost" as a few unconnected nodes do not affect the system validity and it is easy to satisfy through a uniformly random deployment.
- We assume that the message between sensor nodes are authenticated. This is the assumption that has been made in the literature. It can be waived through a key management. This assumption essentially means that we only consider the eavesdropping attack. Strictly, this belongs to the formulation of the adversarial model. But we put here to remind the readers this restriction on the adversarial power.

2.3 Location Privacy

In this subsection, we consider the privacy of a SLP-WSN system. Toward this, we first specify feasible adversarial behaviors and then define the location privacy.

Adversarial behaviors. The adversary has ν devices to perform eavesdropping attacks in the network, each of which has a hearing range h. For any selected position, he can place an eavesdropping device. Any signal transmitted in its hearing range will be captured. For each captured signal, the attacker can localize its immediate sender node.

Remark 1.

1) *In this work, we only consider the eavesdropping attack above. An active attack such as a man-in-the-middle attack is not considered as a convention of known SLP-WSN systems. This can be amended through a key distribution protocol to allow any two neighboring nodes to share a key with which the authentication and confidentiality can be achieved. This is obviously out of the scope of this work and we will not explore this.*

2) *Some works in the literature (e.g., [20, 22, 29]) considered the global eavesdropping attack, where an adversary can eavesdrop any message sent in the network. We feel this attack is too strong and unnecessary. To secure against such an attack, the system must sacrifice the efficiency. For example, the systems in [20, 22, 29] emit many faking packets in order to mislead the attacker. This results in a large energy consumption and is not desired.*

Location privacy. The location privacy is to require that an attacker can not find the physical location of S by performing an eavesdropping attack above. The location privacy is quantified by the number of source messages that S can send before it is localized by the attacker and we call it *SafetyPeriod*. Certainly, *SafetyPeriod* is expected to be as large as possible for better privacy.

2.4 Efficiency Measures

In this subsection, we define three measures to evaluate the efficiency of a SLP-WSN system.

PathRatio. We use *PathRatio* to denote the ratio of the number of edges on the real path that a source message will travel from S to v_0, relative to the number of edges on the shortest path from S to v_0. Note that a small *PathRatio* indicates a small transmission delay and hence is desired.

EnergyRatio. We use *EnergyRatio* to denote the ratio of the total length of messages that nodes in the whole network have sent in order to transmit one source message from S to v_0, relative to the length of messages sent by the nodes when S routes the message only through the shortest path to v_0. This measure is concerned with the energy consumption of the network and hence is better to be as small as possible. If S sends m to v_0 by simply flooding it

into the network, then *EnergyRatio* will be n/d_{Sv_0}, where d_{Sv_0} is the shortest path length from S to v_0 and n is the network size.

DeliveryRate. The *DeliveryRate* is the percentage of the source messages from S that have been successfully delivered to v_0. Usually, for a useful protocol, *DeliveryRate* should be almost 100%.

3 Construction

In this section, we present a new SLP-WSN protocol. In our protocol, $v_3, \cdots, v_{n-1}$ are *ordinary sensors* while v_1, v_2 are *reference sensors*. Reference sensors are functionally identical to ordinary sensors, except that they will, together with the *base station* v_0, play as anchors to help all sensors determine their locations. We present our protocol in stages.

3.1 Deployment

In this stage, a system manager deploys the nodes in a region of radius R as follows.

- Randomly deploy $\{v_3, \cdots, v_{n-1}\}$ in the desired area (of radius R) and place v_0, v_1, v_2 on the perimeter such that any two of them are separated by an angle $2\pi/3$ (as in Figure 1).

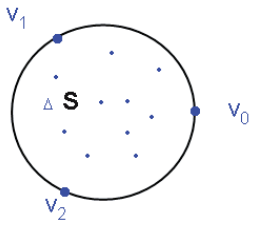


Figure 1: The placement of v_0, v_1, v_2

3.2 Preprocessing

In this stage, each node will obtain some state information for its future execution. This includes its neighboring information, the shortest path to v_0 and the minimum distances to v_0, v_1, v_2, where the minimum distance between v and v' is the minimal number of nodes to traverse from v to v'.

Let $\mathcal{N}(v)$ be the set of nodes within the hearing range of node v and the hearing range of a node is r_0. So, $v_i \in \mathcal{N}(v_j)$ if and only if $v_j \in \mathcal{N}(v_i)$. Clearly, $\mathcal{N}(v)$ is defined only after the deployment.

The preprocessing stage has two sub-protocols. The first one helps each v to learn $\mathcal{N}(v)$ while the second one helps v to learn its location information.

Compute $\mathcal{N}(v)$. In this protocol, each v_i broadcasts hello. When a node v_j receives hello from v_i, it means that v_i lies in its hearing range r_0 and so it adds $v_i \in \mathcal{N}(v_j)$. Since every node v broadcasts hello to its neighbors, any v_j can compute $\mathcal{N}(v_j)$. The formal description is as follows.

1) For $i = 0, \cdots, n-1$, v_i broadcasts $v_i|$hello.

loop Upon $v_i|$hello, each v_j adds v_i into $\mathcal{N}(v_j)$.

2) If no more hello is heard, v_j stores $\mathcal{N}(v_j)$.

We remind again that, as other related works, the privacy model in this paper only considers eavesdropping attacks and hence $\mathcal{N}(v)$ above can be correctly computed. However, as remarked in the privacy model, the privacy against active attacks can be achieved easily through a key management scheme.

Compute the location information. Now we show how to help v_i to compute its minimum distances to v_0, v_1, v_2 respectively. If the distance were Euclidean, then they are sufficient to uniquely locate v_i (by elementary mathematics). However, we do not assume a sensor to be equipped with any positioning tool such as GPS, the Euclidean distance is hard to obtain. Instead, we use the length of the shortest path (i.e., the number of edges on the shortest path) between two nodes as a representation for the distance between them and call it *hop distance*. Under this, a node's distance vector (to v_0, v_1, v_2) should approximately represent its relative location in the network[1].

To compute the hop distance from each v_i to $v \in \{v_0, v_1, v_2\}$, we run the one-to-all shortest path algorithm for three times. Let d_{vu} be the hop distance from node v to node u. Then, the three executions of the algorithm will respectively compute $\{d_{v_0v_i}\}_{i=0}^n, \{d_{v_1v_i}\}_{i=0}^n$ and $\{d_{v_2v_i}\}_{i=0}^n$. The protocol is to continuously update d_{vv_j} for $v = v_0, v_1, v_2$ and finally obtain the correct d_{vv_j}. Since the symbols in $\{d_{v_0v_i}\}_i, \{d_{v_1v_i}\}_i$ and $\{d_{v_2v_i}\}_i$ do not overlap, we can present the three executions in parallel. We use $\texttt{pre}(u)$ to record the next node on the shortest path from u to v_0 in computing $\{d_{v_0v_i}\}_i$. Clearly, starting from any node u and iteratively following $\texttt{pre}(\cdot)$, the shortest path to witness d_{v_0u} is given. Note that the shortest path to witness d_{v_1u} or d_{v_2u} (other than hop distances d_{v_1u}, d_{v_2u}) is not interesting to us and hence is not considered. Details are in Figure 2.

[1]It is possible that two neighboring nodes have the same distance vector. However, it is unlikely that two nodes of several hops away still share the same distance vector. Thus, although the distance vector can not accurately localize a node, it certainly approximates its *relative* location very well.

1. Each v_i lets $d_{vv_i} = \infty$ for $v \in \{v_0, v_1, v_2\}$ and $\mathtt{pre}(v_i) = \perp$. Then, $v \in \{v_0, v_1, v_2\}$ updates $d_{vv} = 0$ and sends $v|v|0$ to $\mathcal{N}(v)$.
2. [**loop**] When v_i receives $v_j|v|\ell$ with $v \in \{v_0, v_1, v_2\}$ from $v_j \in \mathcal{N}(v_i)$, it proceeds only if $d_{vv_i} > \ell+1$. In this case, it updates $d_{vv_i} = \ell+1$, sends $v_i|v|d_{vv_i}$ to $\mathcal{N}(v_i)$, and (if $v = v_0$) also updates $\mathtt{pre}(v_i) = v_j$.
3. If no more update is heard, v_i defines $\mathbf{d}_i = (d_{v_0v_i}, d_{v_1v_i}, d_{v_2v_i})$ and broadcasts $v_i|\mathbf{d}_i$ to $\mathcal{N}(v_i)$.
4. v_j keeps $(v_i, \mathbf{d}_i)$ for $v_i \in \mathcal{N}(v_j) \cup \{v_j\}$, and $\mathtt{pre}(v_j)$.

Figure 2. Compute $\mathtt{pre}(v_i), d_{v_0v_i}, d_{v_1v_i}$ and $d_{v_2v_i}$ for each v_i.

At the end of this stage, v_i obtains $\mathbf{d}_i, \mathcal{N}(v_i)$, $\{\mathbf{d}_j\}_{v_j \in \mathcal{N}(v_i)}$ and $\mathsf{pre}(v_i)$ and it keeps them for the use in the next stage. Again, we remind that as the privacy model considers eavesdropping attack only, all the information can be correctly computed.

3.3 Message Transmission

In this stage, we show how S sends message m to v_0 under the help of some sensors. As our concern is to hide the location of S, we can not route m through the shortest path to v_0. Otherwise, an attacker can stay around v_0 to eavesdrop signals and trace back hop-by-hop. This is feasible, as the shortest path from v to v_0 is fixed (recall that $\mathsf{pre}(v_i)$ is fixed after the preprocessing stage) and one signal allows him to trace one step back (hence d_{Sv_0} signals will lead to S).

The idea of our protocol is as follows. Source S first sends m to an adjacent node v_{i_0}. Then, v_{i_0} chooses a random point in the deployed area through sampling a vector $\mathbf{d}$ that represents a random position's hop distances to v_0, v_1, v_2. It then probabilistically chooses a node $v_{i_1} \in \mathcal{N}(v_{i_0})$ according to a certain distribution and requests v_{i_1} to route m to location $\mathbf{d}$. Here the choice of v_{i_1} has a property that a node closer to $\mathbf{d}$ has a better chance to be selected. Upon m, v_{i_1} chooses a node v_{i_2} and requests it to route m to $\mathbf{d}$. This process continues until m reaches a node v_{i_N} close to $\mathbf{d}$. In this case, v_{i_N} routes m to v_0 via the shortest path.

Before proceeding, we introduce or recall some notations. Some of them will not be used until Section 4. But we put them together for an easy reference later.

- $||\mathbf{d} - \mathbf{d}'|| = \sqrt{\sum_{i=0}^{2} |d_i - d'_i|^2}$.
- $\mathbf{d}_j$ is the vector of the hop distance from v_j to v_0, v_1, v_2.
- $\mathsf{ord}_i(v_j)$ is the order of $||\mathbf{d} - \mathbf{d}_j||$ in the decreasingly sorted list of $\{||\mathbf{d} - \mathbf{d}_t||\}_{v_t \in \mathcal{N}(v_i)}$ for a given $\mathbf{d}$, where $v_j \in \mathcal{N}(v_i)$.
- Given a constant $\omega > 0$ and for $v \in \mathcal{N}(v_i)$, let

$$Q_{v_i,\mathbf{d}}(v) = \frac{|\mathcal{N}(v_i)|\omega + \mathsf{ord}_i(v)}{|\mathcal{N}(v_i)|^2(\omega + .5) + .5|\mathcal{N}(v_i)|}. \quad (1)$$

- *elementary triangle formulae*: if a triangle has two sides of lengths ℓ_1, ℓ_2 with angle θ between them, then the third side has a length

$$L(\ell_1, \ell_2, \theta) = (\ell_1^2 + \ell_2^2 - 2\ell_1\ell_2 \cos\theta)^{1/2}. \quad (2)$$

- R is the radius of the deployed area with origin O; r_0 is the hearing of a sensor node; h is the hearing range of an adversarial device.
- A point in the deployed area has polar coordinates (r, θ), with origin O so that v_0 has the coordinates $(R, 0)$. Thus, v_1 is at $(R, \frac{2\pi}{3})$ and v_2 is at $(R, \frac{4\pi}{3})$.
- δ is a small constant (e.g., about 3) and its concrete value is not important.
- ζ is a constant scaling factor (see details at the end of this section).
- The area within radius r_1 of S is called *threat area* and r_1 is called *threat radius*.

Recall that v_i has the following state from the previous stage.

- $\mathcal{N}(v_i)$: the set of neighbors of node v_i.
- $\mathbf{d}_i = (d_{v_0v_i}, d_{v_1v_i}, d_{v_2v_i})$: d_{vu} is the hop distance (i.e., the minimum number of hops) from v to u.
- $\mathsf{pre}(v_i)$: the first node on the shortest path from v_i to v_0.

Notice that $Q_{v_i,\mathbf{d}}(\cdot)$ is a probability distribution over $\mathcal{N}(v_i)$. We will use this distribution to select the next node before approaching $\mathbf{d}$. This selection has the property that a node v_j with $\mathbf{d}_j$ closer to $\mathbf{d}$ will have a larger probability $Q_{v_i,\mathbf{d}}(v_j)$. One might wonder why we can not simply replace $|\mathcal{N}(v_i)|\omega + \mathsf{ord}_i(v_j)$ with $||\mathbf{d} - \mathbf{d}_j||$. In theory, this is possible. But it has a drawback that when v_i is far from $\mathbf{d}$, $||\mathbf{d} - \mathbf{d}_j||$ will remain almost constant when v_j goes over $\mathcal{N}(v_j)$. Thus, $Q_{v_i,\mathbf{d}}(\cdot)$ will be almost uniformly random. Under this, statistics tells us that the routing will remain about S even after a long time. The constant factor ω is used to adjust the "gap" between $Q_{v,\mathbf{d}}$ and a uniformly random distribution. The gap will decrease when ω increases. Given ω, a smaller $|\mathcal{N}(v)|$ implies a larger gap and so we can increase ω to reduce the gap. A more considerate design is to allow a sensor v to choose its own ω (w.r.t. the value $|\mathcal{N}(v)|$). In this work, we just use a global ω for the convenience of analysis.

In our experiment, we found that choosing $v_{i_{j+1}}$ solely according to $Q_{v_i,\mathbf{d}}(\cdot)$ is problematic. For some networks, the routing will be stuck in a small area for a long time. To avoid this, we make a special rule such that if v_{i_j}

1. If S wishes to send m to v_0, it sends m to an adjacent node v_{i_0}. Upon m, the latter chooses $\theta \in [0, 2\pi)$ uniformly randomly and $r \in [0, R]$ with density $P(r) = \frac{2r}{R^2}$. Next, it computes $\mathbf{d}$ with $d_u = \zeta \cdot L(r, R, \frac{2u\pi}{3} - \theta)/r_0$ for $u = 0, 1, 2$, and samples v_{i_1} from $\mathcal{N}(v_{i_0})$ w.r.t. $Q_{v_{i_0},\mathbf{d}}(\cdot)$ and *special rule* below. Finally, it sends $0|v_{i_0}|m|\mathbf{d}$ to v_{i_1}.
2. [**loop**] (*randomized routing*) If v_{i_j} receives $0|v_{i_{j-1}}|m|\mathbf{d}$, it does the following. If $||\mathbf{d} - \mathbf{d}_{i_j}|| \leq \delta$ (small constant), then it moves to step 3; otherwise, it samples $v_{i_{j+1}}$ w.r.t. $Q_{v_{i_j},\mathbf{d}}(\cdot)$ and *special rule* below, and sends $0|v_{i_j}|m|\mathbf{d}$ to $v_{i_{j+1}}$.
3. [**loop**] (*deterministic routing*) Upon $1|v_{i_{j-1}}|m|$ or when transferred from step 2, if $v_{i_j} = v_0$, then m arrives at v_0 successfully; otherwise, v_{i_j} sends $1|v_{i_j}|m$ to $v_{i_{j+1}} = \texttt{pre}(v_{i_j})$.

Figure 3. Message transmission. *Special rule:* The next sensor policy is amended such that if the current node v_{i_j} has been visited three times, then the next node $v_{i_{j+1}}$ will be the node among $\mathcal{N}(v_{i_j})$ with the minimal distance to $\mathbf{d}$ and that if the current node v_{i_j} has been visited seven times, then v_{i_j} moves to step 3 to route m via the shortest path to v_0.

was visited three times, then the next node $v_{i_{j+1}}$ will be chosen as the node in $\mathcal{N}(v_{i_j})$ closest to $\mathbf{d}$. Under this, the message will not linger around v_{i_j}. Besides this, we also found that there exists a certain bad $\mathbf{d}$ so that no v_i lies in the distance δ to $\mathbf{d}$. In this case, the message will move around $\mathbf{d}$ forever. To avoid this problem, we also make the special rule such that if v_{i_j} has been visited seven times, then it directly routes m to v_0 via the shortest path. Of course, here the threshold three and seven can be modified to other values but we found they work well in our experiments.

For a point at (r, θ), we can see that their *Euclidean distances* to v_0, v_1, v_2 are respectively $L(R, r, \frac{2u\pi}{3} - \theta), u = 0, 1, 2$. However, what we need is a measure that is compatible with the hop distance vectors $\mathbf{d}_i$'s. Fortunately, we find in the experiment that, for a randomly chosen (r, θ), if we scale the Euclidean distance by a factor ζ/r_0, then this scaled distance vector will well approximate a hop distance vector. Further, this ζ only depends on the average node degree $\mathbf{E}(|\mathcal{N}(v)|)$ (e.g., $\zeta = 1.478$ if the average node degree is 7). Intuitively, ζ is affected by two factors: (1) the average hop length in a shortest path is smaller than the full hop length r_0; (2) the shortest path between two nodes is not in a straight line. Both factors will result in ζ larger than 1.

With the above discussion, we are now ready to present our protocol; see Figure 3.

4 Analysis

In this section, we analyze the location privacy of our protocol and discuss its efficiency. When necessary, please see Section 3.3 to recall some notations.

4.1 Location Privacy

Now we consider the location privacy of source S. Before our analysis, we give a sample path for the randomized routing to get a picture of what it looks like; see Figure 4.

We will analyze the following attack framework. The attacker has ν eavesdropping devices. He can place his

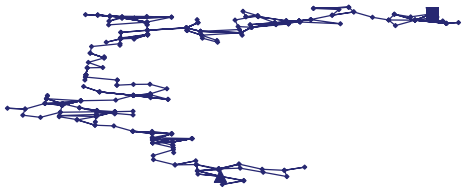


Figure 4: A sample path from source ▲ to random intermediate point ■, with $R = 63r_0$ and $\mathbf{E}[|\mathcal{N}(v)|] = 7$

devices at any location for eavesdropping. We assume that if the attacker captures a message within the radius r_1 of S, then he can localize S. We call this area *threat area* and r_1 the *threat radius*. To properly capture the capability of a real adversary, r_1 can not be large (a large r_1 also implies a strong model and we can not result in an interesting result). We suggest $r_1 = 8r_0$ although our analysis does not depend this. Toward discovering the threat area, an attacker can try any strategy to place his eavesdropping device. Our effort in this section is to find the number of source messages that S can send before the attacker can discover the threat area. This is, we derive the *SafetyPeriod*. This is done in several steps.

For convenience, we call the path from v_{i_0} to v_{i_N} *Path I* and the path from v_{i_N} to v_0 *Path II*.

An eavesdropping attack on Path II is useless. Since Path II starts at v_{i_N}, an eavesdropping attack on Path II can at most trace back to v_{i_N}. This is useless as $\mathbf{d}$ is the uniformly random in the deployed area (independent of the source S).

In the following, we only consider the eavesdropping attack on Path I. We first show that catching the message from v_{i_j} to $v_{i_{j+1}}$ can not imply any information about the location of $v_{i_{j-1}}$ (although it indeed indicates the location of v_{i_j} as assumed).

Under this, one eavesdropped message only exposes one edge on the transmission path. Then, we analyze the edges exposed by eavesdropped messages and derive the probability that an attacker can discover the threat area. From this, *SafetyPeriod* will

be computed.

Tracing back more than one edge from one eavesdropped message on Path I is impossible.
In this part, we argue that an attacker can not trace back more than one edge from one eavesdropped message on the routing Path I. We will not directly prove this rigorously. Instead, we compare our next sensor policy with the uniformly random policy, in which the next sensor node is chosen uniformly randomly from the neighbors of the current node[2]. In contrast, our policy is to choose the next sensor $v_{i_{j+1}}$ according to $Q_{v_{i_j},\mathbf{d}}(\cdot)$ and *special rule* (see Figure 3). Since the next node of the uniformly random policy is independent of the current and previous nodes, tracing back more than one edge is obviously impossible. On the other hand, the next sensor policy through the shortest path to $\mathbf{d}$ is deterministic and hence is the worst. The strategy for proving our policy is good is to find a measurement under which, our policy is close to the uniformly random policy while the shortest path policy is the worst.

Toward this, we define μ_j to be the angle between $\overrightarrow{v_{i_0}v_{i_j}}$ and the average vector $\mathbf{E}(\overrightarrow{v_{i_j}v_{i_{j+1}}})$, where $\mathbf{E}(\cdot)$ is taken over the distribution of $v_{i_{j+1}}$ (for fixed $\mathbf{d}$ and v_{i_j}). We also define θ_j to the angle between $\overrightarrow{v_{i_0}v_{i_j}}$ and $\overrightarrow{v_{i_j}v_{i_{j+1}}}$. If the next sensor policy is good, then θ_j should vary a lot around μ_j. For the uniformly random policy, no matter what μ_j is, θ_j is uniformly random over $[0,\pi]$. In contrast, for the shortest path policy, $\theta_j \equiv \mu_j$. Thus, we consider $\Delta = \mathbf{E}(|\theta_j - \mu_j|)$ as the measurement for the performance of the next sensor policy, where $\mathbf{E}(\cdot)$ is over the distribution of $v_{i_j}, v_{i_{j+1}}, \mathbf{d}$ and μ_j (note that μ_j depends on v_{i_j} and $\mathbf{d}$). Note that Δ varies with j. We run simulations to see how Δ in our scheme performs. We take $R = 63r_0$, $\omega = 1$, and the average node degree $\mathbf{E}[|\mathcal{N}(v)|] = 7$. The result for (Δ, j) is shown in Figure 5, where we notice that the shortest path policy has $\Delta = 0$ while the uniformly random policy has[3] $\Delta = \pi/3$. From the experimental result, we can see that our Δ is very close to that of the uniformly random policy and hence demonstrates its excellent performance against the tracing-back attack for more than one edge.

The probability to discover the threat area.
In the above, we have demonstrated that one eavesdropped message can only expose the underlying edge (in a routing path). Now if an attacker can capture many messages, then their underlying edges

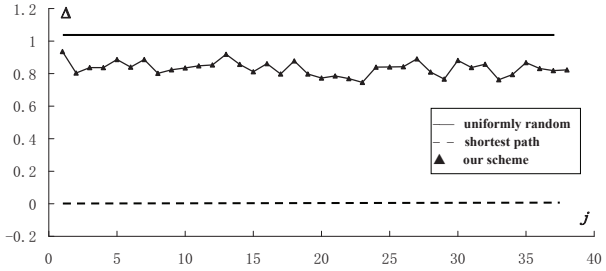


Figure 5: $\Delta = \mathbf{E}(|\theta_j - \mu_j|)$ varies with j: our scheme vs uniformly random policy (constant $\pi/3$) vs the shortest path policy (constant 0).

in the routing paths are exposed. If one of these messages lies in the threat area, then the attacker may realize it and compromise the location privacy of S. In the following, we will calculate the probability that an attacker can catch one message in the threat area when S sends one message. Then, we will use it to compute the *SafetyPeriod*.

If a device is placed on the radius r of S, the probability that an outgoing message from S will be eavesdropped, can be calculated as follows. If $r \le h$, then certainly the message will be catched as the hearing range of the device covers S. If $r > h$, it is easy to see that the device covers the angle (centered at S) of at most $2\beta = 2\arcsin(h/r)$; see Figure 6. Hence, the message is catched with probability at most $\frac{\arcsin(h/r)}{\pi}$.

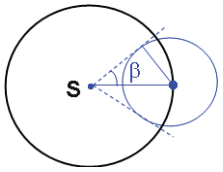


Figure 6: Hearing coverage of an eavesdropping device with respect to S

Now since S is uniformly random in the deployed area, a particular eavesdropping device is on the radius r to S with the probability density $\frac{2r}{R^2}$ (here for simplicity, we assume the maximum distance is still R). Thus, the probability that a single device is placed within the threat radius $r_1 (> h)$ to S and also catches a message, is

$$\begin{aligned}
&\int_0^h \frac{2r}{R^2}dr + \int_h^{r_1} \frac{2r}{R^2}\frac{\arcsin(h/r)}{\pi}dr \\
=&\frac{r^2 * \arcsin(h/r) + h\sqrt{r^2-h^2}}{R^2\pi}\Big|_h^{r_1} + \frac{h^2}{R^2} \text{ (use Maple)} \\
=&\frac{r_1^2 * \arcsin(h/r_1) + h\sqrt{r_1^2-h^2}}{R^2\pi} + \frac{h^2}{2R^2} \qquad (3)
\end{aligned}$$

We can assume $h \le 0.3r_1$, under which, it is well approximated that $\arcsin(h/r_1) \approx h/r_1$ and

[2]Note since a uniformly random policy does not allow a message to go far, this strategy actually is not recommended in the literature. However, since it obviously prevents an attacker from tracing back more than one edge, it is an ideal standard to evaluate the performance of our next sensor policy.

[3]Note that under the uniformly random policy, θ_j is uniformly random over $[0,\pi]$, while μ_j is uniformly random over $[0,\pi]$ (over the distribution of v_{i_j} and $\mathbf{d}$). Thus, $\Delta = \pi/3$, which can be easily calculated from $\mathbf{E}(|\theta_j - \mu_j|)$.

$\sqrt{r_1^2 - h^2} \approx r_1$. Then, Eq. (3) is approximately $\frac{2r_1h}{R^2\pi} + \frac{h^2}{2R^2}$. This is the probability for one device. Since the attacker has ν devices, the probability he catches a message within radius r_1 of S (when one source message is sent) is

$$\frac{2\nu r_1 h}{R^2\pi} + \frac{\nu h^2}{2R^2}. \quad (4)$$

Calculating SafetyPeriod.

Now we calculate the *SafetyPeriod*. If S sends W messages in the *SafetyPeriod*, then an attacker can identify the *threat area* of S only if he can catch at least one message within the radius r_1 of S. Thus, to preserve the location privacy of S, we can use Equation (4) to demand the average number of catched messages in the *threat area* to satisfy $\frac{2Wr_1h\nu}{R^2\pi} + \frac{Wh^2\nu}{2R^2} < 1$. This implies $W < \frac{R^2}{0.5\nu h^2 + 0.636\nu r_1 h}$. Note any W satisfies this restriction will be good. Thus,

$$SafetyPeriod = \lceil \frac{R^2}{0.5\nu h^2 + 0.636\nu r_1 h} \rceil - 1. \quad (5)$$

Take $r_1 = 8r_0, h = 3r_0$. We have *SafetyPeriod*= $\lceil 0.0506R^2/(r_0^2\nu) \rceil - 1$. If $R = 63r_0$ and $\nu = 1$, then *SafetyPeriod*=200. Note we take $\nu = 1$ as the literature does not consider the case $\nu > 1$ and this will be convenient for us to compare. The curve $W = 0.0506R^2/r_0^2$ is shown in Figure 7 with *SafetyPeriod* being the integer value just below the curve.

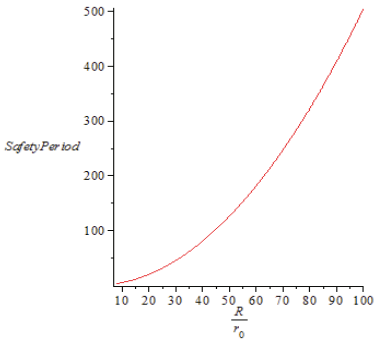


Figure 7: SafetyPeriod vs R/r_0, with $\nu = 1, r_1 = 8r_0, h = 3r_0$. *SafetyPeriod* for R/r_0 is the integer just below the curve.

Remark 2. *The hearing range of adversary h would not be significantly larger than the sensor hearing range r_0, as the signal by the sensor will die out quickly beyond the range of r_0 (or the noise will be more powerful than the signal), under which the adversarial device can not decode correctly (even if it is good). Thus, our sample choice $h = 3r_0$ is reasonable.*

4.2 Efficiency

In this subsection, we discuss the *PathRatio*, *EnergyRatio* and *DeliveryRatio* of our scheme. For the definitions of these measures, see Section 2.4.

In our assumption at Section 2.2, we assume that the network is almost fully connected and thus the message *DeliveryRate* is almost 100%. Our *PathRatio* = *EnergyRatio* also has a good performance. The experimental result is shown in Figure 8, where we depicted the *PathRatio* vs ω for $R = 63r_0$ and average node degree 7. We also depicted the experimental result *PathRatio* vs the average node degree *deg* for $R = 63r_0$ in Figure 9 (where a node degree vs Δ_{18} is also shown). We can see that our *PathRatio* is typically very small while the privacy is preserved well (in Figure 7). Although our experiment is done on average node degree 7, we prefer a smaller degree, because the smaller degree gives a smaller *PathRatio* and *EnergyRatio*, as seen in Figure 9(a). The only problem is that we need to satisfy the almost full connectivity. If there is a strategy to satisfy this for a smaller degree, it is certainly good for our application[4]. Since it is obviously out of the scope of this work, we will not explore this.

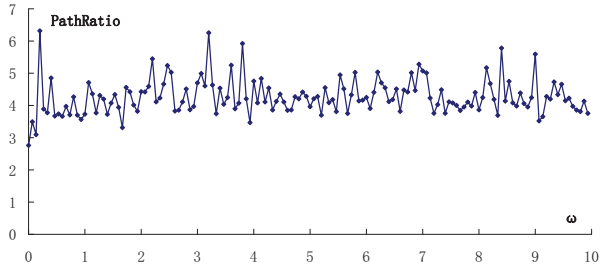
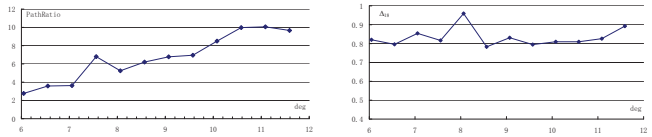


Figure 8: *PathRatio* vs ω ($R = 63r_0$ and average degree $\mathbf{E}[|\mathcal{N}(v)|] = 7$).



(a) node degree vs PathRatio (b) node degree vs Δ_{18}

Figure 9: The effect of node degree to PathRatio and Δ, where $R = 63r_0$ is fixed. We can see that PathRatio increases significantly with node degree while it does not impact Δ significantly.

4.3 Comparison

In this subsection, we compare our scheme with related works and the summary of the comparison appears in Table 4.3. We believe that our criteria and result are satisfactory.

All the schemes [8, 12, 13, 14, 15, 16, 21] use the two-stage routing strategy as we do. We assume the network

[4]Indeed, *SafetyPeriod* is not significantly affected by a reduced degree, as our *SafetyPeriod* analysis in Section 4.1 depends on the average node degree only through Δ while Δ does not depend significantly on this average degree: Δ_{18} does not change significantly with the average node degree (see Figure 9(b)) and Δ_j does not change significantly with j (see Figure 5).

is almost fully connected and so all these works (including ours) have almost 100% *DeliveryRate*.

Flooding-based methods were proposed in [8, 21]. They all have a large energy ratio (which is 125 in their sample experiment) although their *PathRatio* is 1. Since our *PathRatio* can be made small (Figure 8 and Figure 9(a)) as long as the average node degree is not large, our protocol is certainly advantageous to theirs.

Phantom single path method in [8] starts the first phase routing with ℓ steps of directed random walk. However, the directional information is visible and as observed in [15], this reduces the attack complexity by a factor of 2^{ℓ}. This places a concern on their *SafetyPeriod*. The directed random walk was designed as a weak version of a purely random walk, where the latter has the problem that the message will only move nearby the source node. Our method is to approximate the random walk while it moves toward the random intermediate position. This is achieved by carefully choosing the distribution function for the next-hop sensor.

In comparison with [12, 13, 14, 15, 16], our obvious advantage is to remove a node's awareness of its personal location. This is important as it might need an extra hardware such as GPS to realize. Since our *EnergyRatio* and *PathRatio* can be made small as mentioned above, our gain is interesting. In addition, [12, 15] used a mixing ring to hide the routing and thus the ring nodes have a fast power drainage (although authors suggested a leverage strategy, the effect is limited). Lightfoot *et al.* [16] directly routes the message to a random intermediate point deterministically in the first stage and results in a smaller *PathRatio*. As seen before, a deterministic routing is certainly not advantageous as our probabilistic routing. Li and Ren [13, 14] proposed several schemes. The most interesting one (in our view) is the multi-intermediate method. We do not have an obvious advantage over their method other than removing a node's awareness of its own position. For *EnergyRatio* and *PathRatio* in [13, 14], the comparison with us depends on their choice of the number of intermediate nodes.

As a summary, we can safely conclude that our protocol in comparison with [8, 21] has either a much smaller *EnergyRatio* or a better privacy and in comparison with Li *et al.* [12, 13, 14, 15, 16] has the advantage of removing a node's awareness of its own position. As our *EnergyRatio* and *PathRatio* can be made small (with a moderate average node degree), our advantages are interesting.

Table 1: Performance comparison (undesired result marked double black).

	Path Ratio	Energy Ratio	own loction Awareness	Safety Period
SinglePath Phantom [8]	small	small	partial	**small**
Flood [21, 8]	1	**large**	not required	large
[13, 12] [15, 14, 16]	vary	vary	**required**	large
ours	small	small	not required	large

5 Other Issues

In this section, we discuss some other issues that are important for a useful SLP-WSN system.

5.1 Localizing S from a Base Station

Our SLP-WSN system is to prevent an attacker from localizing the source S. However, in some situations, the base station (operated by a personnel) might wish to localize S. In this case, S can send $\mathbf{d}_{i_0}$ of node v_{i_0} to v_0 through our routing system (whenever he moves to a new location). Of course, to be secure, this should be encrypted and authenticated using a secret key shared between S and the base station. When the base station receives $\mathbf{d}_{i_0}$, he can find S by moving toward $\mathbf{d}_{i_0}$. For this to work, he might need to query a node v_j (on his way to S) for its distance vector $\mathbf{d}_j$.

5.2 Service Availability

Network robustness. Network is robust if the network is connected when a small fraction of nodes are out of order. Thus, it is significant to design a robust network. However, since this issue is common in a general wireless sensor network. We will not discuss it here and assume that the network remains almost fully connected even if a small fraction of nodes are out of order.

Routing information update. In our system, we rely on each node and their neighbors' information about the shortest path to v_0 and distance vectors $\mathbf{d}_i$'s. If all nodes are alive and located at the fixed locations, then this information will remain unchanged. If a few nodes die out, then the system can be maintained to continue functioning. Toward this, the system can use a standard network routing update strategy to maintain a node's internal state. Since the number of broken sensors is small, this will be a small workload only. We may also try to keep it working without an update. The only concern is $\mathsf{Pred}(\cdot)$, where when $\mathsf{Pred}(v)$ dies out, our system does not specify what is the next sensor for v. To patch this, v can send the message to the node v_j with the second smallest d_{j0} among its neighbors $v_j \in \mathcal{N}(v)$. If this v_j still dies out, then it can try v_t with the third smallest d_{t0}. It continues until the message is sent out. If the message can not send out at a node v, then the path to v_0 is broken and the delivery fails. If only a few nodes are broken in the network, this bad event should not occur with a noticeable probability. Now if the system runs for a longer period, then many nodes might move or die out. In this case, the system

needs a significant work to update. It might be better to run the preprocessing stage once again. If this update does not occur frequently, we believe that the re-execution is a feasible solution.

6 Conclusion

In this work, we studied the source location privacy in wireless sensor networks, where the source S wants to route a message m to the base station v_0 while keeping its own location private. We proposed a new scheme for this problem. Our scheme routes m through a single path to v_0. Our idea is to first route m to an intermediate position $\mathbf{d}$ while the choice of the next sensor is randomized according to a well-designed probabilistic strategy. Our strategy has the property that a node close to $\mathbf{d}$ has a better chance to be selected. When m approaches $\mathbf{d}$, the underlying node then routes it directly to v_0 through a shortest path. Our protocol performs well at the energy consumption, delivery rate, time delay and the safety period. Importantly, we do not assume that a node is equipped with a localization tool such as GPS. Our scheme has significant advantages over existing protocols.

References

[1] A. Abbasi, A. Khonsari, and M. S. Talebi, "Source location anonymity for sensor networks," in *Proceedings of the 6th IEEE Consumer Communications and Networking Conference (CCNC'09)*, pp. 1–5, 2009.

[2] J. Deng, R. Han, and S. Mishra, "Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks," in *Proceedings of the DSN*, pp. 637, 2004.

[3] M. Dong, K. Ota, and A. Liu, "Preserving source-location privacy through redundant fog loop for wireless sensor networks," in *13th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC'15)*, pp. 1835–1842, 2015.

[4] C. Gu, M. Bradbury, and A. Jhumka, "Phantom walkabouts in wireless sensor networks," in *Proceedings of the 2017 ACM Symposium on Applied Computing, ser. SAC'17*, pp. 609–617, New York, NY, USA, 2017. ACM.

[5] C. Gu, M. Bradbury, A. Jhumka, and M. Leeke, "Assessing the performance of phantom routing on source location privacy in wireless sensor networks," in *IEEE 21st Pacific Rim International Symposium on Dependable Computing (PRDC'15)*, pp. 99–108, 2015.

[6] A. Jhumka, M. Bradbury, and M. Leeke, "Fake source-based source location privacy in wireless sensor networks," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 12, pp. 2999–3020, 2015.

[7] Y. Jian, S. Chen, Z. Zhang, and L. Zhang, "A novel scheme for protecting receiver's location privacy in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 10, pp. 3769–3779, 2008.

[8] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *Proceedings of the 25th IEEE Distributed Computing Systems (ICDCS'05)*, pp. 599–608, 2005.

[9] J. F. Laikin, M. Bradbury, C. Gu, and M. Leeke, "Towards fake sources for source location privacy in wireless sensor networks with multiple sources," in *IEEE International Conference on Communication Systems (ICCS'16)*, pp. 1–6, 2016.

[10] C. T. Li, M. S. Hwang and Y. P. Chu, "An efficient sensor-to-sensor authenticated path-key establishment scheme for secure communications in wireless sensor networks", *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 8, pp. 2107-2124, Aug. 2009.

[11] S. Li, Y. Xiao, Q. Lin, and Z. Qi, "A novel routing strategy to provide source location privacy in wireless sensor networks," *Wuhan University Journal of Natural Sciences*, vol. 21, no. 4, pp. 298–306, 2016.

[12] Y. Li and J. Ren, "Preserving source-location privacy in wireless sensor networks," in *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'09)*, pp. 1–9, 2009.

[13] Y. Li and J. Ren, "Providing source-location privacy in wireless sensor networks," in *Proceedings of International Conference on Wireless Algorithms, Systems, and Applications (WASA'09)*, pp. 338–347, 2009.

[14] Y. Li and J. Ren, "Source-location privacy through dynamic routing in wireless sensor networks," in *Proceedings of IEEE INFOCOM*, pp. 1–9, 2010.

[15] Y. Li, J. Ren, and J. Wu, "Quantitative measurement and design of source-location privacy schemes for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 7, pp. 1302–1311, 2012.

[16] L. Lightfoot, Y. Li, and J. Ren, "Star: design and quantitative measurement of source-location privacy for wireless sensor networks," *Security and Communication Networks*, vol. 9, no. 3, pp. 220–228, 2016.

[17] X. Luo, X. Ji, and M. Park, "Location privacy against traffic analysis attacks in wireless sensor networks," in *Proceedings of the International Conference on Information Science and Applications (ICISA 2010)*, pp. 1–6, 2010.

[18] T. Maitra, R. Amin, D. Giri, P. D. Srivastava, "An efficient and robust user authentication scheme for hierarchical wireless sensor networks without tamper-proof smart card," *International Journal of Network Security*, vol. 18, no. 3, pp. 553-564, 2016.

[19] A. Proa no, L. Lazos, and M. Krunz, "Traffic decorrelation techniques for countering a global eavesdropper in wsns," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 857–871, 2017.

[20] Y. Ouyang, Z. Le, D. Liu, J. Ford, and F. Makedon, "Source location privacy against laptop-class attacks in sensor networks," in *Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks (SecureComm'08)*, pp. 5:1–5:10, 2008.

[21] C. Ozturk, Y. Zhang, and W. Trappe, "Source-location privacy in energy-constrained sensor network routing," in *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04)*, pp. 88–93, 2004.

[22] M. Shao, Y. Yang, S. Zhu, and G. Cao, "Towards statistically strong source anonymity for sensor networks," in *Proceedings of IEEE 27th Conference on Computer Communications (INFOCOM'08)*, pp. 51–55, 2008.

[23] G. Sharma, S. Bala, A. K. Verma, "An improved RSA-based certificateless signature scheme for wireless sensor networks," *International Journal of Network Security*, vol. 18, no. 1, pp. 82-89, 2016.

[24] R. Singh and M. S. Manu, "An energy efficient grid based static node deployment strategy for wireless sensor networks," *International Journal of Electronics and Information Engineering*, vol. 7, no. 1, pp. 32-40, 2017.

[25] W. Yang and W. Zhu, "Protecting source location privacy in wireless sensor networks with data aggregation," in *Proceedings of the International Conference on Ubiquitous Intelligence and Computing (UIC'10)*, pp. 252–266, 2010.

[26] Y. Yang, R. H. Deng, J. Zhou, and Q. Qiu, "Achieving better privacy protection in wireless sensor networks using trusted computing," in *5th International Conference on Information Security Practice and Experience (ISPEC'09)*, pp. 384–395, 2009.

[27] J. Yao and G. Wen, "Preserving source-location privacy in energy-constrained wireless sensor networks," in *Proceedings of the 28th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS Workshops'08)*, pp. 412–416, 2008.

[28] L. Yao, L. Kang, F. Deng, J. Deng, and G. Wu, "Protecting source - location privacy based on multirings in wireless sensor networks," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 15, pp. 3863–3876, 2015.

[29] Z. Zeng, X. Hu, Z. Chen, and H. Liu, "Preserving source-location privacy in wireless sensor networks against a global eavesdropper," in *Proceeedings of 14th Int'l Conf. Wireless Networks (ICWN'15)*, pp. 183–188, 2015.

Biography

Shaoquan Jiang received the B.S. and M.S. degrees in mathematics from the University of Science and Technology of China, Hefei, China, in 1996 and 1999, respectively. He received the Ph.D degree in Electrical and Computer Engineering from the University of Waterloo, Waterloo, ON, Canada, in 2005. From 1999 to 2000, he was a research assistant at the Institute of Software, Chinese Academy of Sciences, Beijing; from 2005 to 2013, he was a faculty member at the University of Electronic Science and Technology of China, Chengdu, China; from 2013 to now, he is a faculty member at Mianyang Normal University, Mianyang, China. He was a postdoc at the University of Calgary from 2006 to 2008 and a visiting research fellow at Nanyang Technological University from Oct. 2008 to Feb. 2009. His research interests are key stream generators, public-key based secure systems and secure protocols.