Computer Networks 53 (2009) 1512–1529

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

# Privacy-aware routing in sensor networks

Haodong Wang *, Bo Sheng, Qun Li

*Department of Computer Science, College of William and Mary, Williamsburg, VA 23187, USA*

ARTICLE INFO

*Article history:*
Received 21 May 2008
Received in revised form 3 February 2009
Accepted 5 February 2009
Available online 13 February 2009

Responsible Editor: S. Sicari

*Keywords:*
Sensor networks
Location privacy
Security
Traceback

ABSTRACT

A typical sensor network application is to monitor objects, including wildlife, vehicles and events, in which information about an object is periodically sent back to the sink. Many times, the object needs to be protected for security reasons. However, an adversary can detect message flows and trace the message back to its source by moving in the reverse direction of the flows. This paper aims to maximize source location privacy, which is evaluated by the adversary's traceback time, by designing routing protocols that distribute message flows to different routes. First, we give the performance bound for any routing scheme. Then, we present our routing schemes, which maximize the adversary's average traceback time and achieve max–min traceback time given certain energy constraints. We then propose WRS, a suboptimal but practical privacy-aware routing scheme, and provide simulation results. Finally, we extend the discussion to an extreme adversary model, which allows the adversary to deploy an adversary sensor network to monitor the message routing activities. Accordingly, we propose a random schedule scheme to confuse the adversary. To reduce the message delivery time, we give an approximation algorithm for message routing.



## 1. Introduction

Sensor networks will be prevalent in the near future for various applications, including object and event monitoring. A common communication paradigm for sensors is to obtain information about objects or events and send the data back to a base station (or sink) for further analysis. The wireless communication path from the object to the base station may jeopardize the safety of the object if an adversary, who is capable of detecting the message flow, traces back to the message source by moving along the reversed path. The object, e.g., an animal of an endangered species, or a vehicle of military aides, may have to be protected for safety reasons and the related location information should not be disclosed. This concern will become even more serious for future sensor network prevalence in pervasive computing applications, as the ubiquitous information collections doubtlessly encroaches on the privacy of the people involved.

This paper explores the location privacy problem in sensor networks. We aim to hide the location of the message source and make it more difficult for an adversary to trace messages back to the source location. We assume that a security infrastructure, such as secure communication, has already been built in. That is, no information carried in the message (e.g., packet head) will be disclosed, allowing the adversary to gain any knowledge about where the message comes from. The adversary observes the wireless communication within a certain detection range and traces toward the message source by moving, in each step, to the node that transmits the detected target information.

Many message routing protocols have been proposed for sensor networks [1–5]. None of them are designed for location privacy protection. Kamat et al. [6] proposed Phantom routing to solve a similar privacy issue. However, as we will show in Section 7.3, the random-walk-based Phantom routing has poor performance in defending against the adversary's traceback, even if the adversary

* Corresponding author. Tel.: +1 7572213468.
*E-mail addresses:* wanghd@cs.wm.edu (H. Wang), shengbo@cs.wm.edu (B. Sheng), liqun@cs.wm.edu (Q. Li).

1389-1286/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2009.02.002

has very limited traffic monitoring ability. More recently, Metha et al. [7] and Shao et al. [8]propose source location protection schemes under a global traffic analyzer. The two approaches only partially solve the problem. The ConstRate and *k*-anonymity [7] schemes rely on global sensor stimulation and are very resource demanding. FitProbRate [8], however, sacrifices location privacy for short message delivery delay. As we will present in Section 8, our solution minimizes message delay while still achieving the perfect location privacy in the presence of a global attacker.

In this paper, we start the discussion from a simple model where there is only one source node and one adversary, and the adversary always starts the traceback from the sink location. As we will show in Sections 4.1 and 6.4, our theoretical model can also be applied for multiple adversaries and multiple data sources. The time for the adversary to trace back to the source is a natural metric for location privacy. Even if the adversary has limited monitoring power, the adversary can follow any random message path and thus trace back to the message source. We use average traceback time and the possible minimal traceback time it takes for an adversary to reach the source as two metrics for location privacy. Average traceback time signifies an expected performance for location privacy. The minimal traceback time, which shows the worst case scenario, assumes that the adversary has the best luck possible, taking the route with the shortest time to find the source.

We address the location privacy issue under a complete adversary model. When the adversary has limited detecting power, we design routing algorithms to maximize the traceback time. We formulate this problem as an optimization problem constrained by the energy budgets that are allowed for use in message routing. To gain more understanding about this issue, we have tried to look at the problem from different perspectives. First, we give an approximation of the performance bound in a generalized scenario as a guideline for network routing design. Our result indicates that the traceback time is proportional to the number of nodes involved in routing. Given a certain sensor density, the number of nodes participating in message routing indicates the degree of how dispersed in the message routes, which produces longer and more scrambled routing paths that delay the adversary's traceback progress. Then, we show how to optimize the routing performance by considering several special cases in which fixed routes are given. The fixed routes are also categorized as routes that are well separated, without intersection in the middle and splicing routes. Although this seems quite restricted, many applications fit in these constraints. For example, an application may require the routes to be well separated so that the adversary has little chance to capture sufficient messages for message content decryption. In addition, many applications also dictate fixed routes to avoid certain dangerous areas where adversaries gather, or to force the routes to pass through certain points for various reasons such as information multicast or data aggregation.

When the adversary is more powerful, e.g., being capable of deploying a sensor network to monitor the traffic, we propose a random schedule scheme in which each node transmits at a certain time slot in a fixed period such that the adversary would not be able to profile the difference in communication patterns among all the nodes. Obviously, this scheme requires a large number of sensors to participate in the message transmission between the source and the sink, so that only a very small portion of these sensors (which are on the routing path) transmit the valid messages; others just send dummy messages. From the adversary's point of view, the sensors in the whole area are flooding messages and no routing path can be inferred from the communication pattern. As radio communication consumes a significant amount of energy in sensors, our goal is to minimize the message transmission delay so as to keep this "flooding" period as short as possible. There are two ways to reduce the message transmission delay: either increase the data rate or use more routes between the source and the sink. Considering that the message rate at the forwarding nodes cannot be changed (otherwise the adversary would easily identify the message forwarding nodes and then the routing path), the problem of minimizing the message transmitting delay is equivalent in finding as many disjoint routing paths as possible so that more message packets can be routed in parallel. We give an approximation algorithm to find the optimal $k$ disjoint routing paths to deliver the data messages.

To the best of our knowledge, this paper is the first to formulate the location privacy as an optimization problem. This paper aims to build a theoretical foundation for privacy-aware routing in sensor networks. Several papers have worked on different routing schemes for location privacy preservation, but little is known about the theoretical bounds for those schemes. We also show how to mathematically analyze the performance in terms of location privacy. This paper does not consider all schemes for preserving location privacy, but examines only routing protocols in which messages follow predefined routes.

## 2. Related work

Internet anonymity and privacy problems have received extensive attention [9–14]. The location privacy discussed in this paper has two fundamental differences from prior work. First, Internet anonymity relies upon channel secrecy (e.g., secret keys) to protect logical location privacy, while location privacy in this paper addresses the issue of physical location privacy. For example, there is a strong connection between the message header and the identity of the Internet users, while this kind of binding does not exist in wireless sensor networks. Instead, the location of the source sensor node is detected by the radio signal rather than the message content, given the assumption that all messages are encrypted. Second, there is no power constraint for Internet users, but energy is one of the most critical issues in sensor networks. In the Internet, a user may choose any number of proxies [12] or join in a large and geographically diverse crowd [11] to achieve anonymity. On the contrary, the energy budget in sensor networks is extremely constrained.

In [15], Wright et al. described the *predecessor* attack and the *setup* attack that are effective against various ano-

nymity schemes, including Crowds [11], DC-Net [10], Onion routing [12] and MIX-net [9]. Similarly, their proposed attacking techniques rely on message content analysis, except for multiple collaborating adversaries and timing analysis. As indicated previously, we do not consider this type of attack since we assume that proper encryption has already been applied to the message content (including packet header) so that no content information is revealed. As we will show, our discussion and proposed schemes do address the multiple-adversary problem and timing analysis attack. In particular, our analysis of optimum routing schemes under the adversary model with limited detecting power is also valid when there are multiple adversaries conducting traceback simultaneously, and so is our proposed random schedule scheme when our sensor field is globally monitored by an adversary sensor network. To defend against the timing analysis threat under the global adversary model, our random schedule scheme is designed to hide the real message routing path and therefore defeat the adversary's timing analysis attack.

Much work has been done in providing security to sensor networks [16–21]. However, encrypted message content cannot defeat the adversary's traffic analysis and traceback to the source object location.

Several papers [22–27] discussed privacy and anonymity issues in wireless communications, and propose solutions by manipulating the message contents. The approaches proposed in [22–24] either encrypt or modify the message content (data cloaking) to confuse the adversary and achieve privacy. The Mist Routers [25] offered both location privacy and anonymous communication in ubiquitous computing environments by combining a hierarchical mixed network and a message encryption scheme. In comparison, Jiang et al. [26] and Fu et al. [27] address the privacy issue from the traffic analysis perspective. Jiang et al. [26] proposed a cover mode to keep the protected message flow indistinguishable from the rest of the traffic. Fu et al. [27] designed a digital filtering technology to defeat the flow marking attacks that could degrade anonymity. In contrast to their schemes, this paper addresses the location privacy threat due to the physical wireless medium that allows the adversary to perform traffic analysis to derive the message flows.

The papers most relevant to our work about privacy in sensor networks are [28,6–8]. Ozturk et al. addressed concern about the originator location privacy [28] in sensor networks. They identified the location privacy issue by using a vivid example *Panda–Hunter Game*, then discussed a possible encryption and routing scheme to prevent the adversary (hunter) from locating the panda. Kamat et al. [6] continued the work and proposed the Phantom routing scheme. Message delivery in Phantom routing is conducted in two phases: First, messages are routed a fixed number of hops by using random walk; Second, after finishing random walking, messages are delivered to the sink by using flooding or single path routing. Compared to the routing scheme (e.g., shortest-path routing) without any privacy protection, Phantom routing can achieve a certain degree of location privacy, even though the performance is not satisfying (as we will show in our simulation results). The drawback of this approach is lacking the intuition of routing strategy. In comparison, this paper presents the theoretical foundation in designing a privacy-aware routing in sensor networks. More recently, Metha et al. [7] and Shao et al. [8] proposed location privacy protection schemes under the presence of a global eavesdropper, the second adversary model considered in this paper. Mehta et al. presented two techniques: periodic collection and source simulation. However, the paper does not present the detailed routing scheme that delivers data to the sink during the collecting period. Meanwhile, the source simulation scheme is limited to applications which the source moving pattern is pre-known. The FitProbRate scheme proposed by Shao et al. greatly shortens the message delay with the price of sacrificing source location privacy. In comparison, we strive for achieving the minimum message delay and perfect location privacy at the same time under the presence of a global eavesdropper.

For our extended adversary model, we use node disjoint $k$ minimal weight paths, which has been discussed in [29–34].

## 3. Network and adversary model

We consider a wireless sensor network consisting of sensor nodes that are uniformly and randomly scattered in a sensor field. Each node has the capabilities to collect data and route data to the sink in a multihop fashion. In this paper, we assume sensor nodes are evenly distributed in the sensor field and do not move after being deployed.

We consider two types of adversary models in this paper. First, we focus on the single-adversary model. It will be shown in the next section that the (limited) multiple-adversary model still obeys the general performance of our adversary model. Once an adversary gets close to the source, the source will be disclosed. This may not be true in all cases, but in many scenarios the adversary is capable of detecting the source by other means (other than eavesdropping) within a certain range. We describe the adversary's radio detection model as follows. The adversary may carry a portable or car based Radio Direction Finder [35]. This type of device is normally equipped with two or multiple separate antennas. As shown in Fig. 1, the adversary has two antennas $A1$ and $A2$. Upon receiving radio signal from the antennas, the adversary can easily triangulate on the transmitter. It is also very possible that two or more adversaries work together. By applying current sensor node localization techniques, they can easily

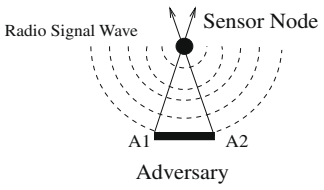



**Fig. 1.** Adversary's radio detection model: The portable or car-based Radio Direction Finder is equipped with multiple antennas, shown as $A1$ and $A2$. With multiple separate receivers, the adversary can easily use triangulation to locate the transmitting sensor node.

pin-point the location of the transmitter. Once detecting a message signal, the adversary quickly moves to the transmitter's location and starts the next message detecting. By repeating this procedure, the adversary can trace back on the message routing path and finally locate the source node. In this paper, we assume the adversary's radio detection is always successful and correct. Second, we extend our discussion to more powerful adversaries. In the worst case, the adversaries may deploy a similar sensor network to monitor every activity at every location. Under such situation, any routing scheme proposed for the first adversary model will fail to protect location privacy because the source sensor node activity will be immediately detected by the adversary's deployed sensors.

Many routing schemes are constrained by energy consumption. We use a very simple energy consumption model: each transmission of a message (i.e., a packet) by a node costs one unit of energy. The energy consumption for receiving and the node's sleep/wake-up schedule can be carefully considered to fit into this model. We omit this detail due to space constraints. In the rest of the paper, the number of messages sent in total and the energy consumption are all normalized. We assume each data packet has enough space to carry one message. In this case, the amount of consumed energy for a message is equal to the path length. Thus we use energy and path length interchangeably.

We model network routes in a directed graph. An edge $(A, B)$ exists if and only if $AB$ is a valid link in one of the routes. Our goal is to assign message flow to all the links (the route segments) so that the traceback time can be maximized. After the message flow is assigned, the routing becomes simple: each node randomly picks a downstream node for message relay according to the flow distribution. In the rest of the paper, except when specified, all of the routing schemes follow this message distribution model.

## 4. Performance bound analysis

Given a sensor network, we are interested in finding the ultimate location privacy we can achieve. In this section, we first develop the performance bound under the assumption that the adversary has the same radio detection range as the sensors' transmission range. Then, we relax the constraints of the adversary's model and allow the adversary to trace back more than one hop each time. Finally, we present our simulation results from our discrete event-based simulations. The performance bound is an approximation of the adversary traceback time; it is by no means an accurate result.

### 4.1. Performance bound for general routing schemes

To study the performance bound of general routing schemes, we consider a sensor field with randomly and evenly distributed $N$ nodes participating in message routing. Let $Freq(i)$ be the frequency of messages seen at sensor node $i$. We denote $L$ as the average routing path length, and normalize the sensor node's transmission range to 1. Therefore, $L$ is actually the number of hops between the source node and the sink, averaged over all routes. In this paper, we assume the message rate, $m$, is small enough so that the time interval for sending any two consecutive messages is much larger than the time that it takes the adversary to travel from one node to another. We denote $T_c$ as the traceback time for the adversary to traverse a routing path with $L$ sensors. At node $i$, it takes $\frac{1}{Freq(i)}$ units of time for the adversary to catch the next message. In total, the traceback time is

$$T_c = \sum_{i=1}^{L} \frac{1}{Freq(i)}. \tag{1}$$

Note Eq. (1) is very general and can be applied to any routing scenario, including multi-path and random routing. When the routing paths are not evenly distributed, and the messages are not evenly dispersed, it is possible that the adversary traceback time on different routing paths can be different. In that case, Eq. (1) is still valid even though the value of $T_c$ would be different for different paths.

For each message generated from the source node, on average it will be propagated $L$ hops along the path from the source node to the sink. Within a time unit, each of $m$ messages reaches $L$ sensor nodes in the sensor field. On the other hand, the total number of routed messages within a time unit can also be given by $\sum_{i=1}^{N} Freq(i)$. Therefore

$$\sum_{i=1}^{N} Freq(i) = m \cdot L. \tag{2}$$

If the routing paths are evenly distributed in the sensor field, and the source node randomly and uniformly picks a path for each message, the participating sensor nodes have approximately the same message frequency $\overline{Freq}$. Then Eqs. (1) and (2) will become

$$T_c = L/\overline{Freq}, \tag{3}$$

$$N \cdot \overline{Freq} = m \cdot L. \tag{4}$$

Combining Eqs. (3) with (4), we have

$$T_c = N/m. \tag{5}$$

Note that the above results also apply to the multiple-adversary model. Suppose $K$ adversaries collaborate and trace back the messages at the same time. In the best case (for traceback), the adversaries are tracing on $K$ independent routing paths. The traceback is $1/K$ times of that of one adversary. Therefore, the traceback time for multiple adversaries still obeys the general performance of the single-adversary model.

### 4.2. Performance bound analysis

In the previous subsection, we assume the adversary is tracing back one hop each time. Given a longer radio detection ability, the adversary can trace back $h$ hops ($h > 1$) each time. Therefore, Eq. (1) should be rewritten as

$$T_c = \sum_{i=1}^{\lceil L/h \rceil} \frac{1}{Freq(i)}. \tag{6}$$

Combining Eq. (6) with Eq. (4), we have

$$T_c = \frac{N \cdot \lceil L/h \rceil}{m \cdot L} \approx \frac{N}{h \cdot m}. \quad (7)$$

Compared with Eq. (5), Eq. (7) introduces one more factor $h$. The average traceback time is inversely proportion to adversary detection range $h$.

Eqs. (5) and (7) reveal that the adversary's average traceback time is determined by the number of nodes involved, the message rate, and the adversary's detection ability. Considering the message rate and the detection model are relatively stable, the only solution that increases location privacy is to have more sensor nodes involved in message routing, which means the routing paths should be dispersed into a larger area.

### 4.3. Simulation results

We have built a discrete event simulator to study the performance bound of general routing schemes. As shown in Fig. 2, we set up a rectangular sensor field with length of 800 m. The sensor node's transmission range is 20 m. In order to simulate the scenario where each involved sensor node has the same message frequency, we design the simulation scheme as follows. On edge *AB*, we deploy a number of source nodes (the number depends on the length of *AB*) so that the distance between every two consecutive source nodes is 20 m. For example, given the length 80 m in Fig. 2, we deploy three sensor nodes $n_1, n_2$, and $n_3$. Then, we deploy the same number of destination nodes on the other edge *CD*, with the destination nodes paired with different source nodes. For example, $n_1$ and $n_4$ form one pair, $n_2$ and $n_5$ form another pair. For each time unit, we randomly pick a source node on *AB* and send a message to its paired destination node on *CD*. The message routing follows the geographic routing scheme. The adversary can start from any position on *CD*. A traceback procedure ends as soon as the adversary reaches any position on *AB*. In order to change the number of nodes involved in routing, we change the width of the network field with the same node density. In a larger network field, we can use more routes and thus more nodes for routing. We use three possible adversary detection ranges in our simulation: 20 m, 30 m and 40 m.

We present our simulation results in Fig. 3. Instead of using traceback time $T$, we actually use the number of messages, for simplicity and accuracy. Eq. (5) can be rewritten as

$$m \cdot T_c = N. \quad (8)$$

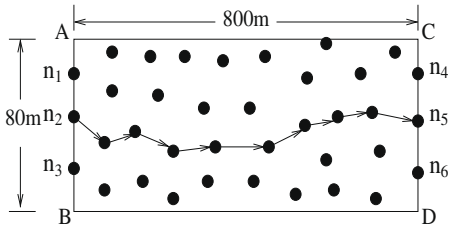



**Fig. 2.** Network setup for performance bound simulation.

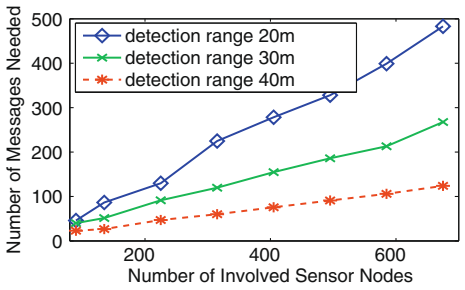



**Fig. 3.** The adversary's traceback time vs. the number of sensor nodes under three different adversary detection ranges.

$m \cdot T_c$ in the left hand side of Eq. (8) is the number of messages the adversary needs in order to reach the source node. Fig. 3 shows that the adversary's traceback time grows linearly with the increasing number of involved sensor nodes under all three different detection models. Moreover, the slope for the detection range of 40 m is approximately twice the slope for the detection range of 20 m, which also matches Eq. (7).

## 5. Average traceback time

We have given the approximate performance estimation for any routing scheme, but how to design a routing strategy to maximize the traceback time is still a question. In this section and the next section, we explore the optimal routing strategies under two different performance metrics: average traceback time and minimal traceback time. This section presents the optimal routing scheme that maximize the average traceback time. We assume the routes are well separated so that there is no transmission interference between any node pair from any two routes, and that the adversary tracing on one route is not able to detect the messages on another route. We start from a simple example with two routing paths. Then, we generalize the problem with $n$ routes.

Suppose we have the routing scenario shown in Fig. 4. Source node $s_k$ has the choice to send messages to either of two routing paths with length[1] $l_1$ and $l_2$ (from now on, we use $l_1$ and $l_2$ to represent the two paths, respectively). Suppose $s_k$ chooses $l_1$ with probability $p_1$, and chooses $l_2$ with probability $p_2$ ($p_1 + p_2 = 1$). Paths $l_1$ and $l_2$ intersect at point $A$, where the adversary is located. Once the adversary starts tracing on one routing path, she will not be able to detect the message on the other path. Therefore, the adversary traceback time along $l_1$ is $l_1/p_1$. Similarly, the traceback time along $l_2$ is $l_2/p_2$. Starting from point $A$, the adversary has probability $p_1$ to get a message coming from $l_1$ and probability $p_2$ to get a message coming from $l_2$. The adversary's average traceback time, $T_a$, can be given by

$$T_a = p_1 \cdot \frac{l_1}{p_1} + p_2 \cdot \frac{l_2}{p_2} = l_1 + l_2. \quad (9)$$

[1] By length we mean the number of hops on that route.

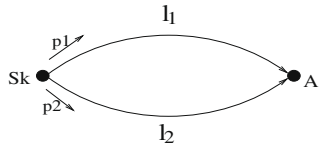



**Fig. 4.** Message distribution scheme with only two paths.

Let $E$ be the amount of energy required to deliver a message from the source to the sink. We assume that the two routes can be chosen from a range of routes with length between $l_0$ and $l_m$ $(E \leqslant l_m)$. Given the following constraints:

$$\begin{aligned} &l_m \geqslant l_1, \quad l_2 \geqslant l_0, \\ &p_1 + p_2 = 1, \\ &p_1 l_1 + p_2 l_2 \leqslant E \leqslant l_m, \end{aligned} \tag{10}$$

the average traceback time $T_a$ is maximized when $l_1 + l_2$ achieves its largest possible value. Without loss of generality, we assume $l_1 \leqslant l_2$. To maximize $l_1 + l_2$, we first increase $l_2$. Notice that the largest possible value of $l_2$ is $l_m$, and $l_1 \leqslant \frac{E - p_2 l_2}{p_1}$, so we have

$$T_a = l_1 + l_2 \leqslant \frac{E + l_m(2p_1 - 1)}{p_1} = \frac{E - l_m}{p_1} + 2l_m. \tag{11}$$

Since $E - l_m \leqslant 0$, the maximum value of $T_a$ is achieved when $p_1 = 1$. Therefore, $Max(T_a) = E + l_m$. Note that the value of $Max(T_a)$ cannot be reached unless $l_m = E$. The reason is that if $p_1 = 1$, then $p_2 = 0$, and we cannot use Eq. (9) to calculate traceback time. Instead, the traceback time $T_a = l_1/p_1 = l_1$.

Now, let us consider the routing scenario with $n$ paths. The average traceback time $T_a = l_1 + l_2 + \cdots + l_n$, and our goal is to maximize $l_1 + l_2 + \cdots + l_n$. We still assume that each path can choose a length between $l_0$ and $l_m$ $(E \leqslant l_m)$.

**Theorem 1.** *Given n routing paths $(l_1, l_2, \ldots, l_n)$ connecting the source node $s_k$ and point A, messages can be routed from $s_k$ to point A through any of the paths. Suppose that these n routes do not intersect at anywhere except at point A. The adversary can then detect the message from any path at point A. Once the adversary starts the traceback procedure on one of the n paths, she cannot detect the message signal from the other paths. Let $P = \{p_1, p_2, \ldots, p_n\}$ be the message probability distribution on $\{l_1, l_2, \ldots, l_n\}$ (note $p_1 + p_2 + \cdots + p_n = 1$). Therefore, the adversary's average traceback time $T_a = l_1 + l_2 + \cdots + l_n$. If we have the following energy constraints:*

$$\begin{aligned} &l_0 \leqslant l_1, l_2, \cdots, l_n \leqslant l_m, \\ &l_1 p_1 + l_2 p_2 + \cdots + l_n p_n \leqslant E, \end{aligned} \tag{12}$$

*the maximum average traceback time $Max(T_a) = (n-1) \cdot l_m + E$.*

**Proof 1.** We can choose $l_2 = l_3 = \cdots = l_{n-1} = l_m$ and $l_1 = E$. Then $T_a = l_1 + l_2 + \cdots + l_n = (n-1) \cdot l_m + E$. This can be achieved by distributing all of the flow to $l_1$ and assigning message probability 0 to $l_2, l_3, \ldots, l_n$. The average traceback time is maximized because there must exist a path with length no greater than $E$ (which is $l_1$), and all other paths have the maximal length. □

Now, let us consider another variation of the problem. Suppose we have $n$ fixed routes with fixed length $l_1 \leqslant l_2 \leqslant \cdots \leqslant l_n$, and the adversary chooses any path with equal probability $1/n$, which is the case when the adversary starts its tracing from a random point in the middle of the network. The best strategy for distributing the message flows is to assign probability 1 to $l_1$ and probability 0 to all other routes, which makes the average traceback time $T_a = (l_1/p_1 + l_2/p_2 + \cdots + l_n/p_n)/n$ to be infinity.

The above analysis states that many routes have to be left unused or used very rarely to maximize the average traceback time. This is true if the adversary does not change position and always waits for the next message on the previous selected traceback path. However, the adversary is normally smarter. Instead of remaining static at one point and waiting for the next message, the adversary may roam around to discover other traceback routes which carry messages more frequently. In case the adversary finds the route that is assigned for message routing with probability 1, the traceback time would immediately be increased to $T_a = l_1$. Therefore, we believe the average traceback time cannot characterize the real scenario. In the next section, we propose a more realistic performance metric: minimal traceback time.

## 6. Max–min traceback time

In the previous section, we have seen that the average traceback time leads to an unreasonable solution and could not characterize the real scenario. Here we propose another more realistic performance metric for location privacy: minimal traceback time, which captures the worst case scenario. Routing schemes with good performance in terms of the average traceback time may perform poorly in the worst case. For example, consider the optimal routing scheme for average traceback time described in the previous section. In the worst case, the adversary may pick the shortest routing path with length $l_1 = E$ and message probability $p_1 \approx 1$. The adversary's minimum traceback time is $l_1/p_1 \approx E$. Thus, in the worst case, the optimal scheme performs no better than a single routing path with the length of $E$.

In the following, we first consider the message routes that are well separated so that they have no common node other than source and sink, then we investigate the splicing routes that are tangled together. For well-separated routes, we consider which routing scheme is optimal given energy consumption constraints. We look at two scenarios: a route can take an arbitrary length and a set of fixed routes, and we find the optimal message flow distribution for them. In the splicing route case, we also look at a set of fixed routes to see how to distribute flows.

### 6.1. Max–min traceback time for length-adjustable routes

In order to maximize the adversary's minimum traceback time, we should avoid following two situations: (1) the majority of messages are routed through minority routes; (2) one or several routing path lengths are significantly shorter than the rest of the routing paths. Given

the same power constraints as in Eq. (12), we arrange the $n$ routing paths in the way shown in Fig. 5. All routing paths are parallel with each other without any intersection between $s_k$ and $A$. Since the length of routing paths is adjustable, we let $l_1 = l_2 = \cdots = l_n = E$. The source node $s_k$ randomly and uniformly distributes the messages to these $n$ routes. Obviously, the adversary's traceback time on all $n$ routing paths is $nE$. Therefore, the adversary's minimum traceback time under this routing scheme is $nE$. Now, we show that $nE$ is the max–min traceback time.

**Theorem 2.** *Given n routing paths $(l_1, l_2, \ldots, l_n)$ connecting the source node $s_k$ and point A, messages can be routed from $s_k$ to point A through any path. Let $P = \{p_1, p_2, \ldots, p_n\}$ be the message probability distribution for paths $\{l_1, l_2, \ldots, l_n\}$ (note $p_1 + p_2 + \cdots + p_n = 1$). If there are the following energy constraints:*

$$\begin{aligned} &l_0 \leqslant l_1, l_2, \ldots, l_n \leqslant l_m, \\ &l_1 p_1 + l_2 p_2 + \cdots + l_n p_n \leqslant E, \end{aligned} \tag{13}$$

*the max–min traceback time $T_{Max-Min} = nE$.*

**Proof 2.** For any routing path distribution $l_i$ and $p_i, 1 \leqslant i \leqslant n$, we want to find $Max\left\{Min\left\{\frac{l_i}{p_i}\right\}\right\}$. Suppose we have the constraints given in (13). Let $a_i = l_i/p_i, 1 \leqslant i \leqslant n$, and the energy constraint can be written as $a_1 p_1^2 + a_2 p_2^2 + \cdots + a_n p_n^2 \leqslant E$. Suppose there is a path $k$ $(1 \leqslant k \leqslant n), a_k = Min(a_i)$. We have $a_1 p_1^2 + a_2 p_2^2 + \cdots + a_n p_n^2 \geqslant a_k p_1^2 + a_k p_2^2 + \cdots + a_k p_n^2 = (p_1^2 + \cdots + p_n^2) a_k$. Therefore, $a_k \leqslant \frac{E}{p_1^2 + \cdots + p_n^2}$. Since $p_1^2 + \cdots + p_n^2 \geqslant \frac{(p_1 + \cdots + p_n)^2}{n} = 1/n$, so $a_k = Min\{l_i/p_i\} \leqslant nE$. Finally, $Max\left\{Min\left\{\frac{l_i}{p_i}\right\}\right\} = nE$. □

### 6.2. Max–min traceback time for length-fixed routes

Suppose there are $n$ fixed routes with length $l_1 \leqslant l_2 \leqslant \cdots \leqslant l_n$. They are well separated from each other so that any pair of routes intersect only at the source and the sink. Our goal is to find the optimal message probability distribution $\{p_1, p_2, \ldots, p_n\}$ that maximizes the adversary's minimum traceback time under the energy constraint $l_1 p_1 + l_2 p_2 + \cdots + l_n p_n \leqslant E$.

As we have discussed in the previous section, for the $n$ routes with the energy constraint $E$, the *max–min* value of the adversary's minimum traceback time is achieved when the traceback time is the same for every path. Likewise, to achieve maximal minimal traceback time, we have to force all the routes to have the same traceback time. If we do not have the energy constraint, a possible solution is to assign the following message distribution:

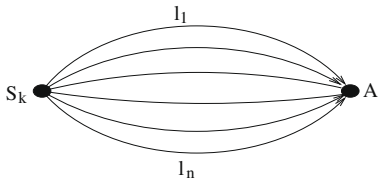



**Fig. 5.** $n$ routing paths are arranged to be parallel with each other.

$p_1 = \frac{l_1}{l_1 + l_2 + \cdots + l_n}, p_2 = \frac{l_2}{l_1 + l_2 + \cdots + l_n}, \ldots, p_n = \frac{l_n}{l_1 + l_2 + \cdots + l_n}$. It is a valid message distribution because $p_1 + p_2 + \cdots + p_n = 1$. Now, the corresponding energy consumption becomes

$$p_1 l_1 + p_2 l_2 + \cdots + p_n l_n = \frac{l_1^2 + l_2^2 + \cdots + l_n^2}{l_1 + l_2 + \cdots + l_n}. \tag{14}$$

Therefore, the solution is feasible when the energy consumption in Eq. (14) is less than or equal to $E$. Obviously, if our energy budget is sufficient (satisfies the above condition), this routing scheme maximizes the adversary's minimum traceback time. This can be explained as follows. The above scheme achieves the same traceback time – $l_1 + l_2 + \cdots + l_n$ on all $n$ routing paths. If we try to increase the traceback time on a specific route $i$, we need to reduce the amount of messages on route $i$. Those messages that originally go through route $i$ should be re-distributed to other routes. Then, the route that gets these extra messages will have a larger message probability. As a result, the corresponding traceback time will be less than the original value. Therefore, the traceback time $l_1 + l_2 + \cdots + l_n$ is the optimal value when $E$ is large enough to cover the routing energy expenditure.

However, since our energy budget is usually tight, which means the value of $E$ is less than the value in Eq. (14), then how do we distribute the messages? Without loss of generality, for a given $E$, assume we can find $k$ such that the first $k$ routes satisfy the energy constraint by using the above routing strategy, but the first $k+1$ routes exceed the energy constraint $E$ by using such a scheme. In mathematical expression, we have

$$\begin{aligned} &\frac{l_1^2 + l_2^2 + \cdots + l_k^2}{l_1 + l_2 + \cdots + l_k} \leqslant E, \\ &\frac{l_1^2 + l_2^2 + \cdots + l_{k+1}^2}{l_1 + l_2 + \cdots + l_{k+1}} > E. \end{aligned} \tag{15}$$

If we only use the first $k$ routes, we can achieve the adversary's minimum traceback time as $l_1 + l_2 + \cdots + l_k$. Notice that we have not used up our energy budget yet, so we can do better because we have not used the rest of the $n - k$ routes yet. Imagine we can move a portion of messages from the first $k$ routes to route $k+1$, so that the traceback time for each of $k$ routes increases at the same rate while the total energy consumption just reaches the value of $E$. If we use $T_s$ to represent the new traceback time for the first $k$ paths, $p_1, p_2, \ldots, p_k$ can be written as $\frac{l_1}{T_s}, \frac{l_2}{T_s}, \ldots, \frac{l_k}{T_s}$, respectively. Therefore, we have

$$\begin{aligned} &\frac{l_1^2 + l_2^2 + \cdots + l_k^2}{T_s} + p_{k+1} l_{k+1} = E, \\ &\frac{l_1 + l_2 + \cdots + l_k}{T_s} + p_{k+1} = 1. \end{aligned} \tag{16}$$

Combining the above equations, we get $T_s = \frac{l_1(l_{k+1} - l_1) + l_2(l_{k+1} - l_2) + \cdots + l_k(l_{k+1} - l_k)}{l_{k+1} - E}$. At this time, $p_{k+1} = \frac{T_s - l_1 - l_2 - \cdots - l_k}{T_s}$, so the adversary's traceback time on route $k+1$ is

$$l_{k+1}/p_{k+1} = \frac{T_s}{(T_s - l_1 - l_2 - \cdots - l_k)/l_{k+1}}. \tag{17}$$

Since $l_1 + l_2 + \cdots + l_k < T_s < l_1 + l_2 + \cdots + l_k + l_{k+1}, T_s - l_1 - l_2 - \cdots - l_k < l_{k+1}$, the adversary's traceback time on route

$k+1$ is longer than $T_s$. Now, we need to prove that $T_s$ is the optimal solution that we can achieve.

**Theorem 3.** *Let $k$ $(0 \leqslant k \leqslant n)$ be an integer such that the following inequalities are satisfied:*

$$\begin{aligned} &\frac{l_1^2+l_2^2+\cdots+l_k^2}{l_1+l_2+\cdots+l_k} \leqslant E, \\ &\frac{l_1^2+l_2^2+\cdots+l_{k+1}^2}{l_1+l_2+\cdots+l_{k+1}} > E. \end{aligned} \tag{18}$$

*Assume* $T_s = \frac{l_1(l_{k+1}-l_1)+l_2(l_{k+1}-l_2)+\cdots+l_k(l_{k+1}-l_k)}{l_{k+1}-E}$. *Then*

$$p_i = \begin{cases} l_i/T_s, & 1 \leqslant i \leqslant k, \\ \frac{T_s-l_1-l_2-\cdots-l_k}{T_s}, & i = k+1, \\ 0, & i > k+1 \end{cases}$$

*gives the optimal message probability distribution on all of the routes.*

**Proof 3.** Assume we have another routing scheme that can achieve a longer value of the adversary's minimum traceback time. Compared with the above scheme, the new scheme should achieve a longer traceback time on *each of* the first $k$ routes. Therefore, the message probability on *each of the* first $k$ routes should be reduced to smaller values, which is equal to "moving" some portion of messages from the first $k$ routes to the rest of the $n-k$ routes. Since we know that the total energy consumption for our proposed scheme is $E$ and in the new scheme we transport message flows from a shorter route to a longer route, the energy consumption for the new scheme will increase, that is, be greater than $E$, which means that the new scheme violates the energy constraint. Therefore, our proposed scheme is the optimal solution with respect to all the constraints. □

### 6.3. Max–min traceback time for splicing network

In many situations, it is not easy to find and deploy well-separated routing paths such as those in Fig. 6 due to sensor field size and the sensor nodes' power constraints. Considering that a long routing path may require a number of remote sensor nodes to participate in the message forwarding task, it is not only a disadvantage in power saving (the operations switching between sleep and active status consumes a lot of power), but also brings about security concerns. Although we disperse our messages into as many routing paths as possible to prevent the possible adversary's traceback, we do want to restrain the messages to a limited area.

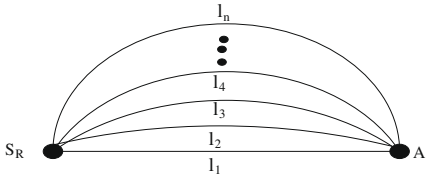



**Fig. 6.** $n$ length-fixed routing paths between $s_k$ and $A$.

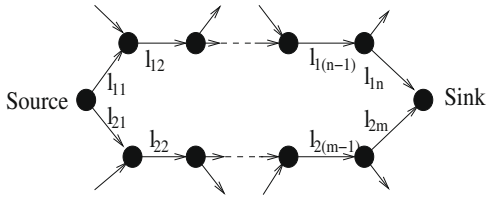



**Fig. 7.** A portion of a splicing network.

A general routing scenario can be shown by a directed graph in Fig. 7. Since we are using splicing network routes, the routing scheme is a little bit different from the previous ones. Each node determines which neighbor to send a message to according to some probability. Our goal is to find the message probability distribution that maximizes the adversary's traceback time in the worst case. As we explained in the previous subsection, the max–min traceback time is achieved when the adversary has the same amount of traceback time on all paths. Note that such an optimum message distribution can be calculated at a centralized node, such as the sink. Since sensor nodes are static, the network topology information can be used to derive the optimum message distributions. Next, we show how to quantitatively determine the message distribution.

As an example, we only focus on two of the routing paths from the source to the sink. Each path is composed of a number of edges. Suppose the upper path (route 1) has $n$ edges, while the lower path (route 2) has $m$ edges. We denote $l_{ij}$ as the length of the $j$th edge of path $i$, $p_{ij}$ as the message probability of the $j$th edge of path $i$. Therefore, the adversary's traceback time on the upper routing path can be written as $\frac{l_{11}}{p_{11}}+\frac{l_{12}}{p_{12}}+\cdots+\frac{l_{1n}}{p_{1n}}$. Similarly, the traceback time for the lower path is $\frac{l_{21}}{p_{21}}+\frac{l_{22}}{p_{22}}+\cdots+\frac{l_{2m}}{p_{2m}}$. Thus, we have the equation

$$\frac{l_{11}}{p_{11}}+\frac{l_{12}}{p_{12}}+\cdots+\frac{l_{1n}}{p_{1n}} = \frac{l_{21}}{p_{21}}+\frac{l_{22}}{p_{22}}+\cdots+\frac{l_{2m}}{p_{2m}}. \tag{19}$$

Since the edges normally do not change after the sensor network is deployed, the values of length $l_{ij}$ are constants. We only need to determine the message probabilities of the edges. Based on the observation that a message routing graph is very similar to multi-loop electric circuits (considering the message flow as the electric currents, and the edge length as the electric voltage), it is natural to apply Kirchhoff's Rules [36] to solve the message probabilities in the routing graph. First, let us define three terms similar to those in the electric circuits, *junction*, *branch* and *loop*.

**Definition 1.** A junction is a sensor node where at least three routing paths meet. The exceptions are the source node and the sink. No matter how many routing paths they are connected to, the source node and the sink are always regarded as junctions.

**Definition 2.** A branch is a routing edge or several serially concatenated edges between two junctions. A branch may consist of several edges because the nodes on the concatenation points are not junctions. In other words, those edges have the same message probability and can be treated as one routing path unit.

**Definition 3.** A loop is composed of two routing paths between a starting junction and an ending junction. Both routing paths begin at the starting junction and end at the ending junction, and they do not intersect at any other junction. Messages can be routed on either path from the starting junction to the ending junction. Each routing path may consist of one or more branches.

Here, our *loop* is different from a conventional "routing loop", which means the situation where a node receives a message which was previously forwarded by itself. We assume a "routing loop" is prevented in our routing protocol and will never happen. In our routing scheme, messages are always moving forward from the source to the destination. For example, the two routing paths in Fig. 7 form a *loop*. Similar to the multi-loop circuit, we can utilize Kirchhoff's Rules to find the message probability for each branch. Here we re-write Kirchhoff's Rules for routing in a splicing network:

> *Kirchhoff's First Rule:* the junction rule. The sum of the message probability coming into a junction is equal to the sum leaving the junction.
> *Kirchhoff's Second Rule:* the loop rule. The adversary's end-to-end traceback time on two paths of a loop is the same.

Based on Kirchhoff's Rules, we can write the junction equations and loop equations by following three steps:

- On the directed routing graph, label the message flow and flow direction in branch.
- Use Kirchhoff's first rule to write down a message probability equation for each junction. In general, if there are $J$ junctions in a routing graph, we need to write $J-1$ junction equations. The equation for the remaining junction is redundant and can be derived from the other $J-1$ equations.
- Use Kirchhoff's second rule to write down loop equations for as many loops as needed to include each branch at least once. To find a loop, we need to pick a starting node and an ending node, then try to find two different paths which both begin and end at these two nodes. At the same time, they do not meet at any third node. When writing the loop equations, we need to make sure equations are independent with respect to each other. A loop equation is guaranteed to be independent as long as there is at least one new branch (that has not previously appeared in other equations) in the loop. In general, if there are $B$ branches and $J$ junctions in a routing graph, in total we will have $B-J+1$ independent loop equations.

Solving the above equations, we can get the optimal message distribution for each path in the splicing network.

### 6.4. Multiple source objects

In the previous two sections, we have explored the optimal routing strategies in a network where there is only one data source and one adversary. In the real world, this kind of network model is rare and restricted. One may wonder whether privacy-aware routing is necessary if there are multiple data sources in the network because the routing messages from multiple data sources that may already confuse the adversary. In this section, we extend our discussion to a network with multiple objects. We explain why multiple data sources cannot confuse the adversary's tracing, so that the location privacy issue is still valid even with multiple source. Our result can also be applied to mobile objects.

Without loss of generality, we start our discussion with two data objects. If the two objects (under sensing) are located far away from each other and their message routing paths do not intersect at all, it is identical to our single source network model and all of our results can be applied. Therefore, we assume the two routing paths intersect at least once as shown in Fig. 8. Suppose that two data sources, $s_1$ and $s_2$, send messages to the sink (where the adversary is located) along the routes $l_1$ and $l_2$, respectively. $l_1$ and $l_2$ intersect at $B$ before they reach the sink.

As discussed in Section 5, if $s_1$ is the only data source, the adversary traceback time to $s_1$ is $l_1$. Similarly, the traceback time to $s_2$ is $l_2$. If $l_1$ and $l_2$ do not intersect, the average traceback time to either $s_1$ or $s_2$ is $\frac{l_1+l_2}{2}$ (assume the data rate is the same). Now we examine whether multiple data sources confuse the adversary's traceback, or increase the traceback time. When two routes intersect at $B$, $l_1$ is divided into $l_{11}$ and $l_{12}$, and $l_2$ is divided into $l_{21}$ and $l_{22}$. Since the data rate from $s_1$ and $s_2$ is the same, the adversary at $A$ has the same probability to detect messages from $l_{12}$ and $l_{22}$. Therefore, the traceback time from $A$ to $B$, denoted at $T_{AB}$, is $\frac{l_{12}+l_{22}}{2}$. Similarly, at point $B$, the adversary has $\frac{1}{2}$ probability of tracing on either $l_{11}$ or $l_{21}$. The expected traceback time for the adversary to reach either of the two data sources from $B$, denoted as $T_{BS}$, is $\frac{l_{11}+l_{21}}{2}$. In total, the expected traceback time to reach either $s_1$ or $s_2$ from $A$ is $T_{AB}+T_{BS}=\frac{l_{12}+l_{22}}{2}+\frac{l_{11}+l_{21}}{2}=\frac{l_1+l_2}{2}$. This result concludes that the adversary's expected traceback time (to reach either source) does not increase when there are two data sources with intersecting routing paths.

We have studied how multiple sources affect the traceback time to any one of the data sources; now let us focus on the traceback time for a specific data source. We still use the routing example in Fig. 8. Without a data source $s_2$, the traceback time to $s_1$ is $l_1$. After $s_2$ is introduced, the average traceback time from $A$ to $s_1$ (suppose the adversary takes the route $l_1 1$ at $B$), denoted as $T_{AS_1}$, changes to $\frac{l_{12}+l_{22}}{2}+l_{11}$. The difference, denoted as $T_{diff}$, can be computed as

$$T_{diff}=l_1-\frac{l_{12}+l_{22}}{2}+l_{11}=\frac{l_{12}-l_{22}}{2}. \tag{20}$$

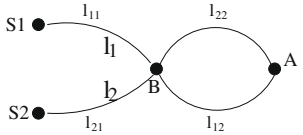



**Fig. 8.** Two data sources.

Therefore, after the second data source is introduced, the change of the traceback time to the first source depends on the difference between $l_{12}$ and $l_{22}$. Note that both $l_{12}$ and $l_{22}$ are routes between $B$ and $A$. Using the general routing schemes in sensor networks, the length of $l_{12}$ and $l_{22}$ should be very close to each other. $T_{diff}$ thus is approximate to 0. Finally, we conclude that multiple data sources do not help confuse the adversary's tracing and increase the traceback time.

## 7. Privacy-aware routing schemes

Inspired by the traceback time analysis for the routing strategies, we discuss two privacy-aware routing schemes in this section. The first routing scheme is called Random Parallel (RP) routing. The strategy is to randomly disperse the source messages into a number of pre-determined parallel routing paths, so that the adversary's traceback progress is deterred due to the fact that the adversary can only perform traceback on a certain routing path. As discussed previously, the pre-determined routing paths are difficult to deploy in a large scale sensor network. Therefore, we propose the second routing scheme, Weighted Random Stride (WRS) routing. WRS routing allows the messages to be routed in a splicing network, which is more practical and natural for sensor networks and requires only a little deployment information.

### 7.1. Random Parallel routing

Random Parallel routing is a straightforward privacy-aware routing scheme which is shown in Fig. 6. Every sensor is pre-assigned $n$ parallel routing paths starting from that sensor and ending at the sink. We assume the arrangement of these $n$ routes satisfies the energy budget. As we discussed in the previous section, the message distribution strategy at the source node is to give the adversary the same traceback time on any routing path. In particular, when the energy budget is large enough, the message probabilities $p_1, p_2, \ldots, p_n$ are arranged in such a way that $l_1/p_1 = l_2/p_2 = \cdots = l_n/p_n$. The adversary traceback time on any path is $l_1/p_1$.

In RP, any two paths should be well separated so that the adversary cannot detect the message transmission on multiple paths at the same time. In practice, the message routing should be restricted to a small area due to the power constraint and security concerns. For simplicity, we use a rectangular routing zone for each sensor. Once the size of the rectangular routing zone is fixed, the number of routing paths and their lengths can be determined. As a result, the message distribution probability for each random parallel path can be determined during the deployment. The main advantage of RP routing is that the messages can be evenly and well dispersed in the designated routing zone to deter the adversary's traceback progress. However, the RP routing method itself reveals the approximate location of the source node to the adversary. Suppose the adversary starts at the sink; he can quickly identify the direction of the source node by only tracing back several messages on any one of the routing paths. Since all routing paths are parallel, the direction of any routing path will lead the adversary to quickly locate the source node. Another disadvantage of RP routing is that each sensor has to have global routing path knowledge because the parallel paths are different for different source nodes.

### 7.2. Weighted Random Stride routing

The intuition of the Weighted Random Stride (WRS) routing scheme is based on the max–min rule in the splicing network, as discussed in the previous section. The goal is to give the adversary the same traceback time on different tracing paths between any two sensor nodes in the network. As we discussed previously, given the network global topology, we can apply Kirchhoff's Rules to derive the message distribution for every routing path. In practice, however, it is very difficult to derive the results for a large scale sensor network due to a number of restrictions. For example, the global topology of sensor locations is very hard to get, and the topology itself also changes a lot over time due to the nature of wireless links. We propose an efficient, light-weight, yet robust WRS scheme to approximately achieve the above goal. The design of the WRS routing scheme considers the fact that sensor network is a splicing network. Instead of distributing the messages to a number of fixed parallel paths as described in RP, WRS scheme allows each individual sensor to make the routing decision locally and independently, with very little deployment information.

To ease the explanation, we use the example shown in Fig. 9 to describe WRS routing. There are two parameters specified in message routing: the forwarding angle and the stride. The forwarding angle is the angle between the projected forwarding route and the line connecting the forwarding node and the sink. When a sensor node $S_1$ transmits a message to the sink (here $S_1$ can be either a source node or an intermediate forwarding node), it first randomly picks a forwarding angle $\alpha$, and selects the neighbor $S_2$ (matching the forwarding angle) as the next hop. The stride is defined as the number of hops associated with the forwarding angle selected by the transmitting node $S_1$. In this example, $S_1$ selects the stride value 3. When $S_2$ receives the message from $S_1$, it notices that the stride is not finished yet, so $S_2$ picks its neighbor $S_3$ as the next hop since $S_3$ fits the forwarding angle. This procedure continues until the message reaches $S_4$. $S_4$ finds that the stride is finished, so it randomly picks another forwarding angle and starts a new stride.

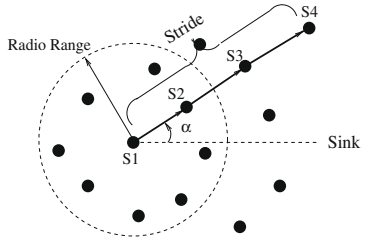



**Fig. 9.** Weighted Random Stride routing scheme.

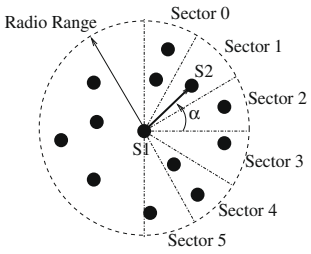



**Fig. 10.** Pick the next hop with weighted probability.

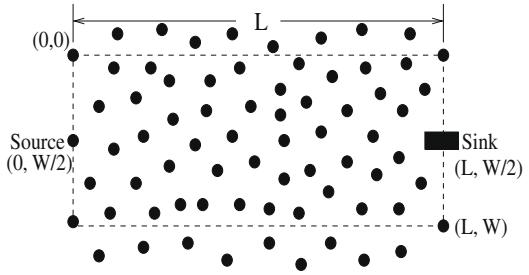



**Fig. 11.** A rectangular routing zone: the length $L$ is the distance between the source node and the sink, $W$ is the width.

It is not difficult to see that a larger forwarding angle leads to a potentially longer routing path. Therefore, different forwarding angles should be picked with different probabilities. In WRS, nodes are arranged to pick a larger forwarding angle with a higher probability. In this way, more messages will be distributed to longer paths so as to deter the adversary's traceback. For practical reasons, we do not require the node to store all forwarding probabilities for every different angle. Instead, we make the following arrangement as shown in Fig. 10 to simplify the procedure. We divide the right half-disc of the node radio coverage (suppose the sink is on the right side, so the node always picks the next hop that is located in the right half-disc) into a number of sectors (six in our example). Now, we randomly pick a sector instead of an angle. Once a sector is picked, the forwarding node selects its neighbor in the corresponding sector that makes the largest forwarding step. Similarly, the probability of selecting the sectors is different. For the example as shown in Fig. 10, sectors 0 and 5 are most likely to be picked, while sectors 2 and 3 have the lowest probability. In our simulation below, the probability of selecting sectors 0 and 1 is three times and twice of that of selecting sector 2, respectively.

### 7.3. Evaluation

To evaluate the proposed the privacy-aware routing schemes, we implement both RP routing and WRS routing in our customized simulator. For the purpose of comparison, we implement a baseline Random Walk (RW) routing scheme which is adopted by Phantom routing [28,6]. In RW routing, the forwarding node randomly and uniformly picks one of its neighbors as the next hop. To make sure the messages will finally reach the sink, each intermediate node always forwards to the neighbor that is closer to the sink.

#### 7.3.1. Simulation setup and metrics

We deploy a large scale sensor network. Sensors are randomly and uniformly distributed in the sensor network. The radio transmission range of the sensor is fixed at 10 m. On average, each sensor has about 20 neighbors. Due to power constraints, message routing should be restricted in a routing zone. As shown in Fig. 11, we assign a rectangular routing zone for the source node. All messages transmitted from the source node should be confined in the rectangular area. The length of the field, $L$, is the distance between the source node and the sink. In the simulation, we fix $L$ to be 800 m. $W$ is the width of the field. The value of $W$ is determined by the energy budget in the network. In the simulation, we change the width from 200 m to 600 m for comparing the performance under different energy budget setups.

Once the width of the routing zone is determined, the routing paths in RP routing can be fixed. In the simulation, we arrange any two adjacent routing paths in RP routing to be separated from each other by 20 m so that the adversary can only trace the message on one routing path as long as his radio detection range is no more than 20 m.

In the simulation, we fix the message rate of the source node at a fixed value, so that we use the number of messages as the metric to measure the adversary traceback performance. We record the number of messages the source node has sent until the adversary successfully locates the source node. There is only one adversary in the simulation. Two radio detection ranges, 10 m and 20 m, are considered.

#### 7.3.2. Simulation results

We perform the first set of adversary traceback simulation, with the adversary detection range of 10 m, for RW, RP and WRS routing, respectively. The routing zone length (the distance between the source node and the sink) is fixed at 800 m. The width is changed from 200 m to 600 m for different energy budget. In the simulation, the adversary always starts tracing from the sink. Once the adversary detects a message transmission, he immediately moves to the location of the transmitting node and waits for the next detection. The traceback ends as soon as the adversary successfully reaches the source node. For each test, the adversary successfully performs traceback for

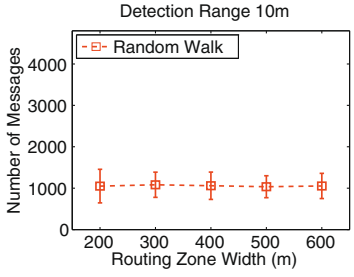



**Fig. 12.** The adversary's traceback time with Random Walk routing, when the detection range is 10 m.

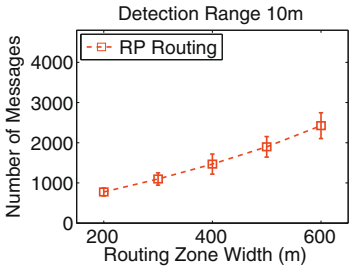



**Fig. 13.** The adversary's traceback time with Random Parallel routing, when the detection range is 10 m.

1000 times. We record the average traceback time (in term of the number of messages) and the standard deviation.

The result of the adversary traceback performance is illustrated in Figs. 12–14, respectively. Fig. 12 clearly shows that the privacy preservation characteristic of RW routing does not change when the routing zone width changes. The adversary traceback time stays around 1000 messages when the routing zone width expands from 200 m to 600 m. This phenomenon indicates that pure random walk routing is independent of the routing zone size. The random walk scheme is not aware of the routing zone change and cannot exploit the extra energy budget to prevent the adversary's traceback.

In comparison, the traceback time in RP routing increases as the routing zone becomes larger. The reason is that the routing paths are well dispersed in RP routing. When the zone size increases, the source node will have more routing paths to which to distribute the messages. Therefore, the adversary has less probability of detecting the message at a specific location, so that the traceback time is longer. Fig. 13 demonstrates that the adversary consistently needs more messages to perform a trace as the routing zone width increases. Given the exact same routing zone width changing from 200 m to 600 m, the adversary traceback time increases linearly from 775 messages to 2424 messages, a much better performance than RW routing.

In WRS routing, we set the stride value to 5. Similarly to Fig. 10, each node has six forwarding sectors, the probability ratio of selecting the forwarding sector is 3:2:1, which means the probability of choosing sectors 0 and 5 is three times more than that for sectors 2 and 3. Differently from the RP routing scheme, WRS allows most of the sensor nodes in the routing zone to participate in the message forwarding. Recall that there are a fixed number of routing paths in RP routing, so the number of participating sensor nodes is limited to those on the routing paths. Therefore, WRS routing yields better traceback time performance than RP routing because the adversary is more confused by many more forwarding sensor nodes from different directions. As we can see in Fig. 14, the adversary has to spend more time to successfully determine the source node location. When the zone width is between 200 m and 500 m, it takes more than twice the traceback time as in RP for the adversary to locate the source node. One may notice that the traceback time decreases when the routing zone width changes from 500 m to 600 m. We call this phenomenon saturation. In our simulation, we find that saturation happens when the zone width is around 500 m. The reason is that the messages cannot reach the additional area when the zone width increases from 500 m to 600 m. In other words, WRS cannot take advantage of the extra energy budget under this situation. We argue that the energy budget is normally very tight so that the chance of saturation is very rare.

In the second set of traceback simulation, the adversary's detection range is doubled to 20 m. Figs. 15–17 illustrate the adversary's traceback performance with 20 m detection range. As we can see, compared to the first set of results, the traceback time in RW and WRS routing reduces more than four times. The reason is that the adversary's effective detection area size increases quadratically when his detection range extends linearly. Interestingly, we find the adversary traceback time in RP routing does not reduce as much as that in RW and WRS. Recall that

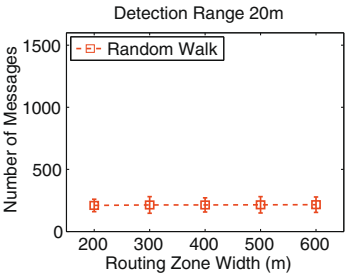



**Fig. 15.** The adversary's traceback time with Random Walk routing, when the detection range is 20 m.

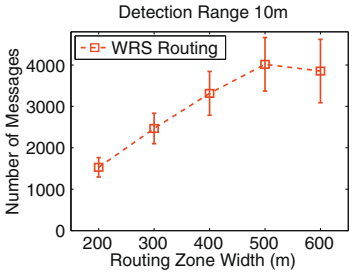



**Fig. 14.** The adversary's traceback time with WRS routing, when the detection range is 10 m.

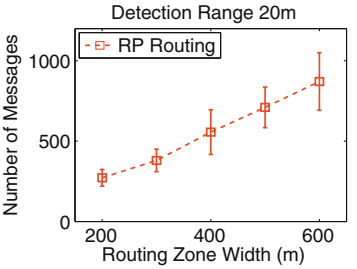



**Fig. 16.** The adversary's traceback time with parallel routing, when the detection range is 20 m.

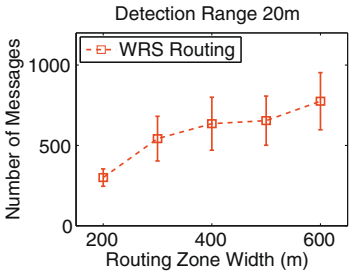



**Fig. 17.** The adversary's traceback time with random routing, when the detection range is 20 m.

we intentionally arrange the routing paths to be separated for approximately 20 m from each other in RP. When the adversary's detection range increases from 10 m to 20 m, the adversary can detect the messages on at most three consecutive paths. That explains why the traceback time in RP reduces by about three times when the adversary detection range becomes 20 m.

As explained in Section 6, many times the minimal traceback time is more critical and practical. Finally, we examine the worst case traceback time for the three routing schemes when the adversary's detection range is 10 m. Among the 1000 adversary's traceback simulation, we pick the fastest traceback and plot the figure shown in Fig. 18. As we can see, RW routing has the worst performance in the worst case. It only takes 570 messages for the adversary to reach the source location. When the routing zone width increases to 600 m from 200 m, this number increases only slightly to 688 messages. Interestingly, RP routing has similar worst case performance as that of RW when the routing zone size is small. However, with the routing zone size enlarged, the worst case traceback time increases quickly. For example, when the width is broadened to 400 m from 200 m, the worst case traceback time increases to 890 from 531 messages. Compared to RW and RP routing, WRS achieves the best worst case performance as expected. When the routing zone width is within 200–500 m, the worst case traceback time increases from 985 messages to 2406 messages, about twice the number of messages in RP. Again, saturation happens when the width becomes 300 m, and the minimum traceback time is moderately reduced to 2287 messages, which is still much higher than RP.

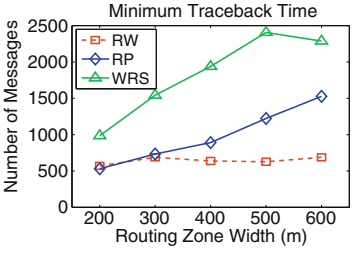



**Fig. 18.** The adversary's minimum traceback time with the detection range of 10 m.

### 7.4. Power consumption overhead

Both the RP and WRS routing protocols improve location privacy by dispersing the messages into different routing paths. Compared with message routing in the greedy shortest-path routing normally used in sensor networks, the messages in RP and WRS travel a longer distance (or more hops) and therefore consume more energy. Now, we investigate the power consumption overhead in both privacy-aware routing schemes.

Since the amount of energy consumption is proportional to the number of hops in the routing path, we denote $C_p = L_p/L$ as the power consumption competitive ratio of the privacy-aware routing scheme to shortest-path routing, where $L$ is the distance (or hop counts) between the source node and the sink, and $L_p$ is the average routing path length in the specific routing scheme, either RW, RP or WRS.

We run the simulation for all three routing schemes: RW, RP and WRS, as well as the shortest-path routing scheme as the base scheme. We continue to use the rectangular sensor field with length of 800 m and the width changing from 200 m to 600 m. In each of above simulation, 1000 messages are routed from the source to the sink, the average number of hops are recorded, and corresponding power consumption competitive ratios are presented in Fig. 19.

It is not a surprise to see that all three privacy routing schemes consume more energy than the base shortest path scheme. What surprises us is that RW has a larger power consumption overhead than RP and WRS, while its anti-traceback performance is much worse (as we discussed previously). The reason can be explained as follows. In RW, each forwarding node equally and randomly selects one of its neighbors (who have a shorter distance to the sink) as the next hop, so the next hop node may not be the one (among the neighbors) that is closest to the sink. As a result, the message forwarding efficiency could be low because it may cost two hops to forward a message which otherwise could be directly routed in just one hop.

Comparatively, the power consumption overhead in RP is very small, just 23% more than the base routing scheme. At the first glance, RP seems more appealing due to the advantage of its low power consumption overhead. However, as we discussed in Section 7.1, RP is not suggested for practical sensor deployment because all routing paths

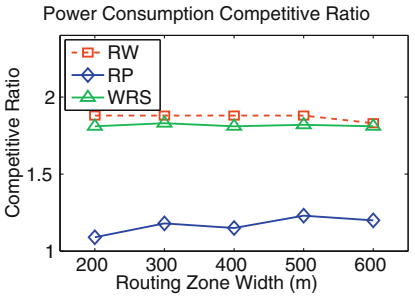



**Fig. 19.** The power consumption comparison among RW, RP and WRS routing schemes.

are parallel with each other, so the routing paths in RP and the corresponding source node location may be easily derived by an adversary after collecting initial network traffic activities.

The WRS scheme, on the other hand, has a larger power consumption overhead and needs around 82% over the base scheme. In fact, the energy overhead of WRS is the trade-off for location privacy. Given the location privacy protection performance of increasing the adversary traceback time from 10 to 40 times (for the corresponding network settings), we believe the approximately 82% energy overhead is a good price for the privacy.

## 8. Adversary sensor network

In this section, we extend our discussion to an extreme adversary model. Instead of placing a certain number of monitoring subjects, the adversary is able to deploy a sensor network to monitor the activities of the sensors in any location in the network. The adversary network is not purposed to detect what our network is monitoring, but it is interested in what assignment our network is involved with and in particular the location of the object that is our network's concern. In this scenario, the adversary is extremely powerful in identifying the monitored object by profiling the network communication activities and analyzing and mining the spatio-temporal relationship among all network communications.

We observe that all of the sensors should transmit their packets at the same rate to prevent the adversary network from detecting any anomaly that may be identified as the data source or the monitored object. Any node (or location) exhibiting more messages in a period encourages close scrutiny and is exposed to a risk of disclosing the monitored object. The solution we propose in this section is to regulate the sensor message transmission rate in a controlled way so that each node (or location) cannot be distinguished by examining the message rate in a period. Each sensor has a scheduled time slot to transmit a fixed amount of messages during a predefined period. In the next period, the sensor will transmit again in the same scheduled time slot. If the sensor has a data message to transmit or relay, it has to wait for its time slot. Otherwise, the sensor still needs to transmit dummy messages if no data messages are available. In this way, all of the sensors have the same message transmission rate in a period. Again, the transmitted messages are all encrypted in a certain way so that the adversary is not able to know the content of any message, but the recipient of the next hop sensor knows a message is destined to it by listening to the message head.

We assume that the clocks on each sensor are well synchronized so that they agree on the message transmission schedule. The scheduled time slot for transmission is a pseudo-random function of the node ID so that each node knows the scheduled transmission slot for any node. Our goal is to design a routing strategy to route messages from the source to the sink with average message delay under the constraints of the controlled transmission schedule. Our algorithms are centralized, assuming that network topology is known to the node who calculates the routing assignment.

### 8.1. Problem

For easy exposition, we assume the data messages are generated at the same time in a bursty fashion. Our algorithm can be easily extended to the case that messages are generated at a certain rate. Our goal is to distribute those messages to the sensors in proximity so that the total delay that those messages go through is minimized. Suppose the source is labeled as "0" and the sink is labeled as "$n$". Strictly speaking, the source is not a sensor, instead it is a conceptual node for easy explanation. The source node connects to the sensors that are in its proximity and can monitor the source for data generation. Since it is a dummy node, we assume the source can send data to the nearby sensors without capacity or rate constraints.

Assume every sensor sends one message per $T$ time units. Let $t_i = f(i)$ be the schedule transmission slot of node $i$, where $t_i \in [0, T)$ and $f$ is a pseudo-random function. Node $i$ will send a message at time $t$ if $t \equiv t_i (\mathrm{mod}\, T)$. We define $d_{ij}$ as the delay at $j$ if $i$ sends a message to $j$ directly

$$d_{ij} = (t_j + T - t_i) \bmod T.$$

The network is modeled as a graph $G(V, E)$, where each edge ($e_{ij} \in E$) connects two nodes ($i, j \in V$) within the communication range. We assign $d_{ij}$ as the weight to edge $e_{ij}$. Let 0 and $n$ be the labels of the source and destination of the messages, respectively

$$d_{0j} = (t_j + T - t_{start}) \bmod T \quad \text{for any edge } e_{0j} \quad \text{and}$$
$$d_{in} = 0 \quad \text{for any edge } e_{in} \text{ connected to the sink,}$$

where $t_{start}$ is the starting time for the source to generate data messages. Our goal is to find routing paths that deliver messages from the source to the sink with the minimum average delay, i.e., the total delay of all messages. It is evident that sending one message with the minimum delay is equivalent to finding the shortest path from the source to the destination in the weighted graph $G$. In the following, we investigate how to route multiple messages.

### 8.2. Multiple messages

If we have $k > 1$ messages to send, one solution is to send all of them through the shortest path. However, due to the schedule constraint, every message arrives at the destination $T$ time later than the previous message. There may exist a more efficient solution, which uses multiple paths, instead of repeatedly using the shortest path, to relay the $k$ messages. A solution $S$ of this problem consists of a set of paths $P = \{p_1, p_2, \ldots, p_m, m \leqslant k\}$ and the corresponding message loads on the paths $M = \{M_1, M_2, \ldots, M_m\}$. In order to avoid message collision, the paths in our solution are node disjoint. Our objective is to minimize the average/total delay of all messages. In other words, our goal is to find a set of disjoint paths and assign message loads to each of them, such that the total delay can be minimized.

Our algorithm is shown in Algorithm 1. We aim to find $L$ node-disjoint routes to transmit messages. During every

time slot of $T$, we inject one message to each of these $L$ routes. Let $l_i$ be the length of route $p_i$. The total delay of this strategy can be expressed as

$$\sum_{i=1}^{L} l_i + \left(\sum_{i=1}^{L} l_i + L\cdot T\right) + \left(\sum_{i=1}^{L} l_i + L\cdot 2T\right) + \cdots = \frac{k}{L}\left(\sum l_i + \frac{kT}{2}\right) - \frac{kT}{2},$$

where $\sum l_i$ is the length summary of all selected paths. Let $SN$ be the set of nodes within communication range of the source, $SN = \{j | e_{0j} \in E\}$. In Algorithm 1, we enumerate all of the possible number of routes in the outer loop, which is upper-bounded by $|SN|$. For each value of $L$, we find a set of $L$ node-disjoint paths, such that $\sum l_i$ is minimized. This problem is equivalent to the minimum $k$ node-disjoint paths problem in graph theory. The existing algorithms, e.g., [29–32], can be applied to our problem. After checking all possible values of $L$, we finally obtain a solution with the minimum total delay, which is stored in variable *opt*.

**Algorithm 1.** Find the optimal solution

```
for L = 1 to |SN| do
    Find L node-disjoint paths that the total length is minimized
    min = total length of L paths
    if (min + kT/2)/L < opt then
        opt = (min + kT/2)/L
        L' = L
    end if
end for
opt = opt · k − kT/2
```

In the following, we show the performance of the approximate algorithm. We use $\{\overline{P}, \overline{M}\}$ to represent our solution, where the route set $\overline{P}$ is obtained by the $k$ node disjoint path algorithm and the message load on each route is the same, i.e., $\overline{M}_i = \frac{k}{L'}$, where $L'$ records the value of $L$ yielding the optimal solution. We use a function $D(P, M)$ to denote the total delay of solution $\{P, M\}$. In our algorithm, $opt = D(\overline{P}, \overline{M})$. Let $\{P^*, M^*\}$ be the optimal solution. In the following, we compare our solution with the optimal one and show *opt* is very close to $D(P^*, M^*)$.

Let $P^* = \{p_1, p_2, \ldots, p_{L_{opt}}\}$ and $M^* = \{M_1, M_2, \ldots, M_{L_{opt}}\}$, where $L_{opt}$ is the number of routes used in the optimal solution. Let $l_i$ be the length of $p_i$. The total delay of $\{P^*, M^*\}$ is

$$\begin{aligned} D(P^*, M^*) &= \sum_{i=1}^{L_{opt}} \sum_{j=1}^{M_i} l_i + (j-1)T \\ &= \sum_i \left(M_i l_i + \frac{M_i(M_i-1)}{2}T\right) \\ &= \sum_i M_i l_i + \frac{T}{2}\sum_i M_i^2 - \frac{kT}{2}. \end{aligned} \tag{21}$$

For each path $p_i$, the delay of the last message is $l_i + T(M_i - 1)$. We can prove the following lemma.

**Lemma 1.** *For any two distinct paths $p_i \in P^*$ and $p_j \in P^*$*

$$|(l_i + T(M_i - 1)) - (l_j + T(M_j - 1))| \leqslant T.$$

**Proof 4.** Proof is omitted due to page limits. □

**Corollary 1.** *For any two distinct paths $p_i \in P^*$ and $p_j \in P^*$*

$$\frac{l_j - l_i}{T} - 1 \leqslant M_i - M_j \leqslant 1 - \frac{l_j - l_i}{T}.$$

Let $p_{min}$ be the path with the minimum length, i.e.,

$$l_{min} \leqslant l_i, \quad i \neq min.$$

Accordingly, the message load on $p_{min}$ is denoted as $M_{min}$. We can prove the following lemma:

**Lemma 2.** $M_{min}$ is the maximum among the optimal message loads of all paths

$$M_{min} \geqslant M_i, \quad i \neq min.$$

**Proof 5.** Proof is omitted due to page limits. □

Now, let us consider another solution, where the route set is the same as $P^*$, but the message load on each route is the same. We use $M\prime$ to indicate this message distribution, i.e., $M'_i = \frac{k}{L_{opt}}$. The following lemma shows the performance of this solution.

**Lemma 3.** *When k is large,*

$$D(P^*, M^*) > \left(1 - \frac{L_{opt}(l_{max} - l_{min})}{kT}\right) D(P^*, M'),$$

*where $l_{max}$ and $l_{min}$ are the longest and shortest path in $P^*$, respectively.*

**Proof 6.** According to Corollary 1, $M_i \leqslant M_{min} - \frac{l_i - l_{min}}{T} - 1$. Recall Eq. (21), the first term is

$$\begin{aligned} \sum_i M_i l_i &\geqslant \sum (M_{min} - 1 - \frac{l_i - l_{min}}{T}) l_i \\ &= (M_{min} - 1)\sum l_i - \sum \frac{l_i - l_{min}}{T} l_i. \end{aligned}$$

Since $M_{min}$ is the maximum message load, it must be greater than the average load $\frac{k}{L_{opt}}$. Therefore,

$$\sum_i M_i l_i \geqslant \frac{k}{L_{opt}}\sum l_i - \frac{l_{max} - l_{min}}{T}\cdot \sum l_i,$$

where $l_{max}$ is the longest path among the path set. Thus,

$$D(P^*, M^*) \geqslant \frac{k}{L_{opt}}\sum l_i - \frac{l_{max} - l_{min}}{T}\sum l_i + \frac{T}{2}\sum_i M_i^2 - \frac{kT}{2}.$$

Since $\sum M_i = k$, we know

$$\sum_i M_i^2 \geqslant \sum \left(\frac{k}{L_{opt}}\right)^2 = \frac{k^2}{L_{opt}}.$$

Therefore, we have

$$\begin{aligned} D(P^*, M^*) &\geqslant \frac{k}{L_{opt}}\sum l_i + \frac{T}{2}\sum_i \left(\frac{k}{L_{opt}}\right)^2 - \frac{kT}{2} - \frac{l_{max} - l_{min}}{T}\sum l_i \\ &= D(P^*, M') - \frac{l_{max} - l_{min}}{T}\sum l_i \end{aligned}$$

$$= D(P^*, M') - \frac{l_{max} - l_{min}}{T}\frac{L_{opt}}{k} \times \left(D(P^*, M') + \frac{kT}{2} - \frac{k^2 T}{2L_{opt}}\right)$$
$$= D(P^*, M') - \frac{l_{max} - l_{min}}{T}\frac{L_{opt}}{k} D(P^*, M') + \frac{l_{max} - l_{min}}{2}(k - L_{opt})$$
$$> \left(1 - \frac{l_{max} - l_{min}}{T}\frac{L_{opt}}{k}\right) D(P^*, M'). \quad \square$$

Recall

$$D(P^*, M') = \sum M'_i l_i + \frac{T}{2}\sum (M'_i)^2 - \frac{kT}{2} = \frac{k}{L_{opt}}\left(\sum l_i + \frac{kT}{2}\right) - \frac{kT}{2},$$

the value of the total delay only depends on $\sum l_i$ and $L_{opt}$. In Algorithm 1, we enumerate all possible values of $L$, which include $L_{opt}$, and try to minimize $\sum l_i$. Thus, *opt* in Algorithm 1 must be no more than $D(P^*, M\prime)$, i.e.,

$$opt \leqslant D(P^*, M') < \frac{kT}{kT - L_{opt}(l_{max} - l_{min})} D(P^*, M^*).$$

Therefore, when $k$ is large, our solution is very close to the optimal solution in terms of total message delay.

### 8.3. Evaluation

To defend against traffic monitoring by the adversary sensor network, all sensors have to transmit messages periodically (in $T$ time units) as long as there is a message to be delivered to the sink. As a result, message delivery becomes a very energy consuming task. Therefore, we want to keep the message delivery time as short as possible. In this subsection, we examine the efficiency of our proposed $L$-disjoint path message delivery solution through simulation.

We set up a rectangular sensor network similar to that presented in Fig. 11, with a length of 800 m and a width of 200 m. Once the sensor network is deployed, the sink can calculate the optimal routing solution as we proposed for each sensor node. We assume each sensor node receives the routing provisioning from the sink, so that there is no processing delay while routing the message from a specific source node to the sink (the routing path is predetermined).

In the simulation, we measure the total amount of time for the source node to successfully deliver various numbers of messages to the sink. Note that the message delivery time here is different from the total delay we discussed in the previous subsection (which is solely for simplifying the analysis). Here, the time is the real world time delay for the source node to deliver the messages to the sink.

We randomly and uniformly deploy 10,000 sensor nodes in the rectangular sensor field. We run the algorithm presented in the previous subsection and find a total of $k = 16$ paths. The length of the 16 paths is shown in Table 1.

Given the 16 routing paths, we estimate the time delay for the source node to deliver various numbers of messages to the sink. For simplicity, we set $T$ to 1 s. Each sensor node is allowed to transmit either a real message or a dummy message in 1 s. For example, as shown in Table 1, the shortest path between the source and the sink is 87 hops. It thus takes 87 s for the source node to transmit one message to the sink. Now, we compare the message delivery time given a different number ($k$) of paths, and plot the results in Fig. 20.

**Table 1**
Length of shortest 16 paths between the source and the sink.

| Path | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Hops | 87 | 88 | 89 | 89 | 89 | 89 | 89 | 89 |
| Path | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Hops | 90 | 90 | 90 | 90 | 91 | 91 | 92 | 93 |

As we can see, when there is only one message to be sent, the message delivery time is the same for different $k$. As the number of messages increases, however, we start to notice a difference in time delay. Considering that we have 10 messages to deliver, if $k = 1$, all messages have to be sent through the only path; it therefore takes nine extra time cycles to delivery 10 messages, for a total of 96 s. If $k = 2$, the source node sends five messages to one of two paths, so the total delay is 92 s. We can get the results for other three cases in a similar fashion.

Interestingly, we notice that the time delay for $k = 16$ is larger than that of $k = 8$ when the number of messages is less than 50. The reason is that, as we can find in Table 1, the longest path length of 16 paths is 93 hops, while the longest path length of eight paths is only 89 hops. As we discussed in the previous section, our algorithm assigns the same message load to each path, so that the longest path in $k = 16$ takes an extra four cycles to deliver a message compared to the longest path in $k = 8$. As a result, the time delay for $k = 16$ is larger when the number of messages is small. The advantage of $k = 16$ starts to show when the number of messages is more than 60.

Overall, multi-pathing does help to reduce the message delivery time, which in turn reduces the energy consumption of the sensor network. However, it does not mean more paths will bring more benefits. If $k$ becomes larger, the longest path length may be very long, which could increase the message delivery time. As shown in Fig. 20, the message delivery time for eight paths and 16 paths is very close. Sixteen paths do not bring significant benefit over eight paths.

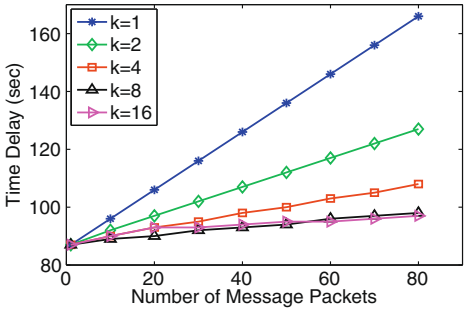



**Fig. 20.** Time delay for delivering various number of messages from the source node to the sink, given a different number of paths.

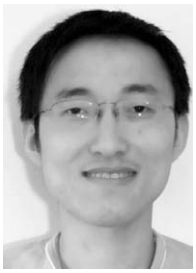
## 9. Conclusion

In this paper, we focus on the location privacy problem in sensor networks. We formulate the problem as an optimization problem in terms of the average traceback time and minimal traceback time for the adversary to reach the message source starting from the sink. We show that the traceback time is related to the number of sensor nodes involved in routing. We give routing strategies to maximize the average and minimal traceback time for a set of fixed routes. Based on it, we propose the WRS, a privacy-aware routing protocol. Our simulation results show that WRS significantly hampers the adversary's traceback progress compared with the Random Walk scheme. We also extend the adversary model to a more powerful one in which an adversary sensor network is deployed to monitor our sensor network communication activities. We show an approximation algorithm to route messages with minimal average delay.

## References


[1] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, in: MOBICOM, Boston, MA, August 2000.
[2] B. Karp, H. Kung, Greedy perimeter stateless routing, in: MOBICOM, 2000.
[3] F. Ye, A. Chen, S. Lu, L. Zhang, A scalable solution to minimum cost forwarding in large sensor networks, in: Tenth International Conference on Computer Communications and Networks, 2001, pp. 304–309.
[4] F. Ye, S. Lu, L. Zhang, Gradient broadcast: a robust, long-live large sensor network, in: Tech. Report, Computer Science Department, UCLA, 2001.
[5] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: 33rd Annual Hawaii International Conference on System Sciences, 2000, pp. 3005–3014.
[6] P. Kamat, Y. Zhang, W. Trappe, C. Ozturk, Enhancing source–location privacy in sensor network routing, in: ICDCS, Columbus, Ohio, June 2005.
[7] K. Metha, D. Liu, M. Wright, Location privacy in sensor networks against a global eavesdropper, in: ICNP, Beijing, China, October 2007.
[8] M. Shao, Y. Yang, S. Zhu, G. Cao, Towards statistically strong source anonymity for sensor networks, in: IEEE INFOCOM, Phoenix, AZ, April 2008.
[9] D. Chaum, Untraceable electronic mail, return addresses and digital pseudonyms, Communications of the ACM (CACM) 24(2) (1981) 84–88.
[10] D. Chaum, The dining cryptographers problem: unconditional sender and receipient untraceability 1(1) (1988) 67–75.
[11] M. Reiter, A. Rubin, Crowds: anonymity for web transaction, in: ACM Transaction on Information and System Security, vol. 1(1), June 1998.
[12] M. Reed, P. Syverson, D. Goldschlag, Anonymous connections and onion routing, in: IEEE JSAC Copyright and Privacy Protection, 1998.
[13] M. Jacobsson, Flash mixing, in: Proceedings of Symposium on Principles of Distributed Computing, May 1999.
[14] I. Goldberg, D. Wagner, E.A. Brewer, Privacy-enhancing technologies for the internet, in: IEEE COMPCON, February 1997.
[15] M. Wright, M. Adler, B. Levine, C. Shields, An analysis of the degradation of anonymous protocols, in: Proceedings of the ISOC Symposium Network and Distributed System Security (NDSS), February 2002, pp. 38–50, outstanding Paper Award. [Online]. Available: <http://prisms.cs.umass.edu/brian/pubs/wright.ndss01.pdf>.
[16] A. Perrig, R. Szewczyk, V. Wen, D. Culler, D. Tygar, Spins: security protocols for sensor networks, ACM/Kluwer Wireless Networks Journal (WINET), September 2002.
[17] C. Karlof, N. Sastry, D. Wagner, Tinysec: a link layer security architecture for wireless sensor networks, in: SENSYS, Baltimore, MD, November 2004.
[18] W. Du, J. Deng, A pairwise key pre-distribution scheme for wireless sensor networks, in: ACM CCS, 2003.
[19] L. Eschenauer, V. Gligor, A key-management scheme for distributed sensor networks, in: ACM CCS, November 2002.
[20] D. Liu, P. Ning, Establishing pairwise keys in distributed sensor networks, in: ACM CCS, Washington, DC, October 2003.
[21] H. Chan, A. Perrig, Pike: peer intermediaries for key establishment in sensor networks, in: INFOCOM, Miami, FL, March 2005.
[22] J. Deng, R. Han, S. Mishra, A performance evaluation of intrusion-tolerant routing in wireless sensor networks, in: IPSN, Palo Alto, California, 2003, pp. 349–364.
[23] M. Gruteser, G. Schelle, A. Jain, R. Han, D. Grunwald, Privacy-aware location sensor networks, in: HotOS IX, 2003.
[24] Y. Zhang, W. Liu, W. Lou, Y. Fang, MASK: anonymous on-demand routing in mobile ad hoc networks, IEEE Transactions on Wireless Communications 5(9) (2006) 2376–2385.
[25] J. Al-Muhtadi, R. Campbell, A. Kapadia, M.D. Mickunas, S. Yi, Routing through the mist: privacy preserving communication in ubiquitous computing environments, July 2002, pp. 65–74.
[26] S. Jiang, N.H. Vaidya, W. Zhao, Routing in packet radio networks to prevent traffic analysis, in: Proceedings of the IEEE Information Assurance and Security Workshop, West Point, NY, July 2000.
[27] X. Fu, Y. Zhu, B. Graham, R. Bettati, W. Zhao, On flow marking attacks in wireless anonymous communication networks, in: ICDCS, 2005, pp. 493–503.
[28] C. Ozturk, Y. Zhang, W. Trappe, Source–location privacy for networks of energy-constrained sensors, in: WSTFEUS, 2004.
[29] J. Suurballe, Disjoint paths in a network, Network 4 (1974) 125–145.
[30] J. Suurballe, R. Tarjan, A quick method for finding shortest pairs of disjoint paths, Network 14 (1984) 325–336.
[31] R. Bhandari, Optimal physical diversity algorithms and survivable networks, in: ISCC, IEEE, Washington, DC, USA, 1997.
[32] A. Srinivas, E. Modiano, Finding minimum energy disjoint paths in wireless ad-hoc networks, Wireless Network 11 (2005) 401–417.
[33] R. Andersen, F. Chung, A. Sen, G. Xue, On disjoint path pairs with wavelength continuity constraint in wdm networks, in: INFOCOM, 2004.
[34] J. Tang, G. Xue, W. Zhang, Interference-aware topology control and qos routing in multi-channel wireless mesh networks, in: ACM MobiHoc, 2005.
[35] HoppyTron.com. Doppler direction finder kit. [Online]. Available: <http://radio_tower_finder.hobbytron.com/r-ddf1.html>.
[36] A.R. Hambley, Electrical Engineering: Principles and Applications, third ed., Prentice-Hall, 2004.


**Haodong Wang** is currently a Ph.D. candidate at Computer Science Department in the College of William and Mary. He got his B.S. from Tsinghua University and M.S. from Penn State University. His research interests are sensor network applications, security and privacy, security schemes on resource constrained devices, and wireless networks.



**Bo Sheng** received his B.S. in Computer Science from Nanjing University, China. He is currently a graduate research assistant in Computer Science Department at College of William and Mary.

**Qun Li** is an assistant professor in the Department of Computer Science at College of William and Mary. He holds a Ph.D. degree in computer science from Dartmouth College. His research interests include wireless networks, sensor networks, RFID, and pervasive computing systems. He received the NSF Career award in 2008.