A Novel Scheme for Protecting Receiver's Location Privacy in Wireless Sensor Networks

Ying Jian, Shigang Chen, Zhan Zhang, and Liang Zhang

***Abstract*—Due to the open nature of a sensor network, it is relatively easy for an adversary to eavesdrop and trace packet movement in the network in order to capture the receiver physically. After studying the adversary's behavior patterns, we present countermeasures to this problem. We propose a location-privacy routing protocol (LPR) that is easy to implement and provides path diversity. Combining with fake packet injection, LPR is able to minimize the traffic direction information that an adversary can retrieve from eavesdropping. By making the directions of both incoming and outgoing traffic at a sensor node uniformly distributed, the new defense system makes it very hard for an adversary to perform analysis on locally gathered information and infer the direction to which the receiver locates. We evaluate our defense system based on three criteria: delivery time, privacy protection strength, and energy cost. The simulation results show that LPR with fake packet injection is capable of providing strong protection for the receiver's location privacy. Under similar energy cost, the safe time of the receiver provided by LPR is much longer than other methods, including Phantom routing [1] and DEFP [2]. The performance of our system can be tuned through a few system parameters that determine the tradeoff between energy cost and the strength of location-privacy protection.**

***Index Terms*—Sensor networks, location privacy.**

I. Introduction

SENSOR network technologies promise drastic enhancement in automatic data collection capabilities through efficient deployment of small sensing devices. A sensor network typically consists of a large number of resource-constrained sensor nodes. Each node acts as an information source, collecting data samples from its environment and transporting data to a receiver via a multi-hop network, in which each node performs the routing function. With the availability of cheap wireless technologies and micro sensing devices, sensor networks are expected to be widely deployed in the near future [3], [4].

The open nature of wireless communication makes it easy for attackers to eavesdrop or inject data packets in a sensor network. Furthermore, unlike other wireless networks composed of mobile devices such as laptops and PDA's with human presence, sensor networks are usually deployed in open areas, where unattended sensor nodes lack physical protection. This means attackers will encounter much fewer obstacles when attacking a sensor network.

Manuscript received February 13, 2007; revised August 13, 2007 and November 6, 2007; accepted November 8, 2007. The associate editor coordinating the review of this paper and approving it for publication was S. Shen. This work is supported in part by US National Science Foundation under career award 0644033.

The authors are with the Department of Computer & Information Science & Engineering, University of Florida (e-mail: {yjian, sgchen, zzhan, lzhang}@cise.ufl.edu).

Digital Object Identifier 10.1109/T-WC.2008.070182

Privacy in sensor networks may be classified into two categories [1]: *content privacy* and *contextual privacy*. Threats against content privacy arise due to the ability of adversaries to observe and manipulate the content of packets sent over a sensor network. This type of threats is countered by encryption and authentication. However, even after strong encryption and authentication mechanisms [5], [6] are applied, wireless communication media still exposes contextual information about the traffic carried in the network. For example, an adversary can deduce sensitive information from a sensor network by eavesdropping the network traffic and analyzing the traffic patterns. In particular, the location information about senders/receivers may be derived based on the direction of wireless communications. In this paper, we focus on the protection of location privacy for the receiver (or the base station) in sensor networks.

It is very important to protect the receiver's location privacy in a sensor network. First, in many sensor networks, the receiver is the most critical node of the whole network, as the responsibility of the receiver (i.e., the base station) is to collect data from all sensors. Since all sensors send data to a single node (the receiver), this creates a single point of failure in the network. A sensor network can be rendered useless by taking down its receiver. Second, in some scenarios, the receiver itself can be highly sensitive. Imagine a sensor network deployed in a battlefield, where the receiver is carried by a soldier. If the location of the receiver is exposed to adversaries, the soldier will be in great danger.

There are several ways that an adversary can trace the location of a receiver. First, an adversary can deduce the location of the receiver by analyzing the traffic rate. This *traffic-analysis attack* is introduced and studied in [2]. The basic idea is that sensors near the receiver forward a greater volume of packets than sensors further away from the receiver. By eavesdropping the packets transmitted at various locations in a sensor network, an adversary is able to compute the traffic densities at these locations, based on which it deduces the location of or the direction to the receiver. However, to perform the traffic-rate analysis, an adversary has to stay at each location long enough such that sufficient data can be gathered for computing the traffic rate. This process takes long time as the adversary moves from location to location. Second, an adversary can reach the receiver by following the movement of packets. This *packet-tracing attack* is first studied in [1], where the sender's location privacy, instead of the receiver's, is considered. In this attack, an equipped adversary can tell the location of the immediate transmitter of an overheard packet, and therefore he is able to perform hop-by-hop trace towards the original data source. We will show that the technique

of packet tracing can be used to locate the receiver as well (Section III). Because the packet-tracing attack does not have to gather traffic-rate information, it allows an adversary to move quickly from location to location towards the receiver. The packet-tracing attack may even be able to trace a mobile receiver due to its fast response, whereas the slow response of the traffic-analysis attack makes it unsuitable for such a task. In this paper, we focus on studying the defense measures against the packet-tracing attack.

When a traditional single-path routing protocol is used, a sensor network is extremely vulnerable to the packet-tracing attack, as the routing paths are fixed and point to the receiver. By eavesdropping the packet transmission, an adversary is able to move one hop along the shortest path towards the receiver for each packet overheard.

In order to protect the receiver's location privacy, we propose a couple of countermeasures against the packet-tracing attack. First, we propose a new location-privacy routing protocol, called *LPR*, to provide path diversity. Second, we combine this routing protocol with *fake packet injection* to minimize the information that an adversary can deduce from the overheard packets about the direction towards the receiver. Under such a protection scheme, an adversary can hardly distinguish between real packets and fake packets, or tell which direction is towards the receiver.

Defending against the packet-tracing attack is a challenging problem. Cryptography does not help because the adversary deduces information simply by overhearing and following the radio transmissions. In order to remove the directional property in the movement of packets destined for a receiver, a considerable number of obfuscating transmissions have to be made. Path diversity provided by LPR inevitably leads to longer routing paths, and transmitting fake packets consumes extra energy. The stronger the protection for the receiver is required, the higher the overhead will be. To address the overhead problem, we design our system in such a way that one can easily tune the tradeoff between the protection strength and the overhead introduced in the network. It should also be noted that, if the security of the receiver is of great importance, overhead may be a price that one has to pay even in sensor networks, when better alternatives do not exist.

The rest of the paper is organized as follows. Section II discusses the related work. Section III defines the problem model. Section IV proposes our new scheme for location privacy protection. Section V presents the analytical results. Section VI evaluates the performance of our defense scheme by simulations. Finally, Section VII draws the conclusion.

II. Related work

Recently, location privacy has gained more and more attention. Different approaches are designed to protect users' privacy in location tracking systems [7], [8], [9], [10], which determine the users' positions for location-based services. Spreitzer et al. [7] make use of a location broker residing at the middleware layer. Hoh et al. [8] create path confusion by crossing paths in areas where at least two users meet. Gruteser et al. [9] perturb the sensed location data to meet the k-anonymity criterion. Al-Muhtadi et al. [10] preserve location privacy through a hierarchy of "mist routers" and a handle-based virtual circuit routing protocol. Location privacy in these studies is content-oriented, where location information is collected and protected as the users' private data.

Onion routing [11] is designed to provide anonymous communications that are resistant against eavesdropping and traffic-analysis attacks on the Internet. Its goal is to hide the identities of the end hosts in a communication session. Designed for the conventional Internet, onion routing employs different network and threat models from the ones suitable for the location-privacy problem in sensor networks. Furthermore, the large communication/computation overhead introduced by onion routing makes it too expensive to be used in sensor networks. MASK [12] deals with passive eavesdropping attacks in mobile ad hoc networks. It conceals the nodes' network/MAC addresses in order to achieve anonymity in communications. But the paper does not specifically consider the packet-tracing attack.

In [2], Deng et al. address the problem of how to hide the location of the base station in a sensor network. A protection method called DEFP (Differential Enforced Fractal Propagation) is proposed in their work, in which techniques of multi-path routing and fake message injection are introduced. However, the work concentrates on the traffic-analysis attack, which determines the base station's location through the measurement of traffic rates at various locations. We have pointed out that the traffic-analysis attack takes longer time to find a receiver than the packet-tracing attack. The simulation results in Section VI will demonstrate that the method in [2] does not perform well in defending against the packet-tracing attack. In [13], Deng et al. propose another technique for protecting the base station against traffic-rate analysis attacks. The transmission times of the packets are randomly delayed in order to hide the traffic pattern and the parent-child relationship under a certain traffic rate model. However, this approach introduces extra delay for delivering packets in a sensor network.

In [1], [14], a routing protocol called *Phantom routing* is designed to protect the location privacy of source nodes (senders) in a sensor network. In Phantom routing, every packet takes a random walk before reaching the sink, which makes it harder for an adversary to trace the movement of packets. However, even with the ability to alter routing paths randomly, Phantom routing can not protect the receiver's location privacy well, because when there are many source nodes in a sensor network, the traffic as a whole still points to the receiver. In [15], a source-sink based random walk is proposed to defend the location privacy of source nodes against a particular type of attacks. This approach cannot protect the receiver, for the same reason that randomized routing alone cannot change the general trend of the traffic as a whole from flowing towards the receiver.

III. Network and Threat Models

A. Network Model

A sensor network consists of a receiver and a number of sensors, deployed in a certain region. Each sensor has a transmission range of r. If the distance between two sensors

is no more than r, they can directly communicate with each other. We do not assume a specific MAC protocol. The link-layer transmission is performed by local broadcast, which is common for wireless medium. *Source nodes* are those sensors that report data to the receiver. Any sensor can become a source node as long as it has something to report to the receiver. We assume that, after a sensor becomes a source node, it periodically sends packets to the receiver for a certain period of time. The receiver has two basic functions: a) broadcasting beacon packets to build the routing structure and b) receiving data packets from all source nodes. Since this paper focuses on contextual privacy, we assume that adversaries can only overhear the packet transmissions but not their actual content. This content protection can be achieved through an encryption method such as [5]. The receiver may move to a new location to receive data. When it is on the move, we assume the receiver is not receiving packets because the existing routing structure still points to its previous stationary location. Obviously, the packet-tracing attack is ineffective when the receiver is mobile. The adversary would trace to the previous stationary location while the receiver has already moved away. Once the receiver arrives at its next stationary location, it broadcasts beacon packets to rebuild the routing structure before receiving data packets.

In order to collect data from the field, a routing protocol is needed for packets to be forwarded from sources to the receiver. Theoretically, a broadcast protocol can be used, in which every data packet is flooded to all nodes in the whole network, including the receiver. Broadcast is extensively used in the route discovery phase of many routing protocols such as AODV [16], and improvement is made in other works [17], [18], [19]. A broadcast protocol is able to achieve location privacy for the receiver because, under broadcast routing, every packet is equally forwarded to all directions and every node in the network "receives" a copy of the packet, which makes it impossible for an adversary to tell which direction points to the receiver. However, broadcast routing has an extremely high energy cost, which renders this approach impractical. Another security problem of broadcast routing is that it quickly exposes the locations of all sensors in a network.

Many routing protocols establish a single path from each source node to the receiver. One of such protocols is described as follows. Each time the receiver moves to a new location, it broadcasts a beacon packet in the network. When a node receives a beacon for the first time, it forwards the beacon to its neighbors by a local broadcast. The beacon roughly follows a shortest-path tree to all sensors, which record their parents as the next hops to the receiver. Data packets will then follow the reverse direction of the broadcast tree towards the receiver. This procedure is similar to the interest propagation phase and the data propagation phase in the directed diffusion scheme [20], where "gradients" from each node towards the receiver are first built before data packets can be routed. As explained in the introduction, single-path routing is vulnerable to the packet-tracing attack.

Because single-path routing is not safe for the receiver and broadcast routing is not practical, a different routing scheme is needed. In this paper, we propose a new routing protocol with fake packet injection to protect the receiver's location privacy. The protocol makes a tunable tradeoff between protection strength and overhead.

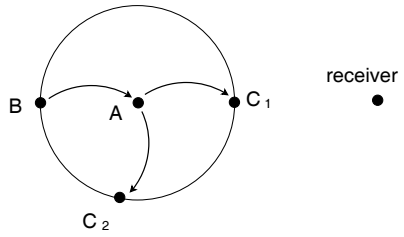


Fig. 1. Behavior of the adversary.

B. Adversary Model

Before studying the problem of location privacy, we first characterize the adversary: what he can do and what he cannot do. We assume the adversary has the following characteristics, some of which are borrowed from the "hunter-panda" game in [1].

- The only goal of the adversary is to capture the receiver. He does not interfere with the proper functioning of the network. He is informed, which means he always knows the protection strategy being used in the system.
- The adversary is equipped with supporting devices, such as antenna and spectrum analyzers, so that he can measure the arrival angle of a packet as well as the strength of the signal. From these two measurements, after he overhears a signal, he is able to estimate the location of the sending node. For the purpose of simplicity, we assume the hearing radius of the adversary is equal to the sensor's transmission range r.
- The adversary can choose to follow a packet, or stay at the same place to capture and analyze more packets. The movement of the adversary is far slower than the movement of a packet in the network.
- The adversary has memory so that he can remember his path and do backtracking. When the adversary stays at a node and cannot hear any packet for a long time, he can choose to step back to his previous locations.
- Finally, the adversary is able to find the receiver visually when he is close enough to the receiver.

C. Packet Tracing

Because a packet is transmitted as a local broadcast, an adversary overhearing the transmission can only tell the location of the immediate transmitter but not the location of the node that is receiving the packet. Let us illustrate how an adversary traces packets in a sensor network by an example in Fig. 1. Suppose the adversary resides at node A. He overhears a transmission made from node B. Shortly after, he overhears a transmission from node A. Then, he overhears a transmission from node C_1, which reveals the location of C_1. For now, we ignore the arrow from A to C_2 in the figure. Based on the above sequence of transmissions, the adversary learns that a packet was sent from B to A and then to C_1. The adversary will move to C_1, hoping that he is one hop closer to the receiver.

In order to camouflage the movement of the real packet, node A may send a fake packet to C_2, as shown in the figure. After overhearing two transmissions from A and then two subsequent transmissions from C_1 and C_2 respectively, the adversary knows that the packet from B to A has been forwarded to either C_1 or C_2, and he has to pick one to trace. In this example, the adversary may guess that the packet sent to C_1 is the real one and the one to C_2 is a fake. The reason is that, with respect to the forwarding line from B to A, the deviation of C_2 from this line is greater than that of C_1. Because the goal of the routing protocols is to deliver a packet to its destination along a path that is as short as possible, the adversary reasonably decides that C_1 has a greater chance to be the real next hop to the receiver. This analysis demonstrates the ability of an adversary to infer the receiver's location through information overheard locally.

There are other types of information analysis. For example, an adversary may stay at one position and keep eavesdropping there for a while. After overhearing a sufficient number of packets, the adversary can determine the neighbor node or the direction that gets the most number of packets, and then he will move to that direction. To provide robust protection for the receiver, we must carefully design our system to resist the adversary from doing these kinds of analysis.

IV. Distributed Solution for Location Privacy

A. Location Privacy Routing (LPR)

Traditional routing cannot protect the receiver's location privacy because all packets from a source node are routed along a fixed (shortest) path towards the receiver. An adversary is able to move one hop closer to the receiver for each packet overheard. We take two steps to remedy this problem. First, we design a location-privacy routing protocol (LPR), which randomizes the routing paths so that the forwarding direction of packets is not always towards the receiver. Statistically, the adversary has to take many more hops before reaching the receiver because he is frequently deviated towards wrong directions due to randomized routing. Second, in the next subsection, we will augment LPR with fake packets such that the probability of forwarding a packet to any neighbor is equalized, which makes the overheard packets useless to the adversary. With these two solutions, if the adversary follows the overheard packets, his path will become a completely random (instead of directed) one. In the following, we describe the protocol of LPR.

To support LPR, each sensor divides its neighbors into two lists: 1) a *closer list*, consisting of neighbors that are closer to the receiver, called *closer neighbors*, and 2) a *further list*, consisting of neighbors that are further (or at equal distance) from the receiver, called *further neighbors*.[1] If every sensor knows its own, the receiver's, and the neighbors' geographic locations, these two lists can be easily constructed based on the Euclidean distances between the nodes. For example, in Fig. 1, node A has two further neighbors, B and C_2, and it has one closer neighbor, c_1. If geographic locations are not known, we can build the two lists as follows. Each time the receiver moves to a new position, it broadcasts a beacon packet in the network. This packet carries a hop count whose initial value is zero. When a sensor receives the beacon for the first time, it increments the hop count in the packet by one, records the hop count, and forwards the packet to its neighbors. After the beacon broadcast completes, neighbors exchange their recorded hop counts, based on which they construct their closer/further lists. We emphasize that the beacon broadcast does not expose the receiver's location. First, it happens only once a time after the receiver gets to a new position. An adversary can only make one movement based on this broadcast.[2] Second, due to the assumption that the packet content is protected by an encryption method, the adversary cannot distinguish between beacon packets and fake packets that will be introduced in the next subsection.

After the closer/further lists are built, LPR works as follows. When a sensor forwards a packet, it randomly selects a neighbor from one of its two lists as the next hop. More specifically, it selects the next hop from the further list with probability P_f, and from the closer list with probability $1-P_f$, where P_f is a system parameter. Because the next hop is randomly selected, the routing path for packets from the same source node to the receiver is randomized, and a packet is not always forwarded towards the receiver.

Let n_c be the number of closer neighbors, n_f be the number of further neighbors, and $\lambda = \frac{n_c}{n_f}$. The probability for a packet to be forwarded to a closer neighbor is $\frac{1-P_f}{n_c}$, and the probability to a further neighbor is $\frac{P_f}{n_f}$. The ratio of these two probabilities is $\frac{1-P_f}{P_f\lambda}$. The following property obviously holds.

Property 1: For any sensor i that is not in the immediate neighborhood of the receiver, under LPR, a) the expected packet rate from i to *any closer neighbor* is the same, b) the expected packet rate from i to *any further neighbor* is also the same, and c) the ratio of these two expected rates is $\frac{1-P_f}{P_f\lambda}$.

On one hand, if the sensors mostly choose their next hops from the closer lists, the routing paths will be short and the energy efficiency will be good. However, the protection for the receiver's location privacy becomes weak because, by following the overheard packets, an adversary can still move quickly towards the receiver. On the other hand, if the sensors frequently choose the next hops from the further lists, the energy efficiency will be lower, but the protection for location privacy will be strengthened because the adversary will be frequently diverted away from the receiver. Therefore, there exists a tradeoff between energy efficiency and location privacy. The tradeoff can be tuned by adjusting the value of P_f.

Suppose the closer/further lists are constructed based on the hop distance to the receiver. Consider a packet that is d hops away from the receiver. Let x_d be the expected number of hops the packet has to travel before reaching the receiver. In

[1]The terms, "closer" and "further", are used with respected to the receiver. For example, when we say that j be a closer neighbor of sensor i and k a further neighbor, we do not mean j is closer to i than k does. We mean that j is closer to the receiver than i does and k is further from the receiver.

[2]Recall that we assume the movement of the adversary is far slower than the movement of packets. When the broadcast "wave" passes the adversary, he has the chance of making one move and, after that, he is behind the front of the broadcast wave.

order for x_d to be finite, P_f has to be smaller than 50%. We have the following difference inequality.

$$x_d \leq 1 + x_{d+1}P_f + x_{d-1}(1 - P_f) \quad (1)$$

After making one hop, with a probability of P_f, the packet is forwarded to a neighbor further away and has to take up to x_{d+1} additional hops on average to reach the receiver. With a probability of $(1 - P_f)$, the packet is forwarded to a neighbor in the closer list and has to take x_{d-1} additional hops on average. Note that the further list includes those neighbors that are at the same distance to the receiver. Therefore, $x_{d+1}P_f$ is an over-estimation for the average additional hops from a neighbor in the further list to the receiver. Consequently, the right side is an upper bound of x_d. Solving the above difference inequality, we have

$$x_d \leq \frac{d}{1 - 2P_f}$$

The expected length of the routing path in LPR is $\frac{1}{1-2P_f}$ times that of the shortest path. Below we discuss the strength and weakness of LPR.

$Strength$: In LPR, the next hop from a sensor to the receiver is not fixed. Sometimes the next hop does not even point to the receiver, which makes it harder to perform the packet-tracing attack. As a routing protocol, LPR guarantees every packet to be delivered to the receiver, as long as $P_f <$ 50%. It is easy to implement and only require one broadcast from the receiver (each time it moves to a new position) to setup the routing structure. It is flexible, allowing an application to tune the tradeoff between energy efficiency and protection strength through a system parameter.

$Weakness$: If we apply LPR alone, the protection for location privacy will not be strong enough because the overall traffic trend in the network still points towards the receiver. Although this problem can be alleviated by setting a higher value for P_f, it will lead to longer delay for packet delivery and higher energy cost. As we have discussed in Section III, the adversary can stay at one location and keep eavesdropping there for a certain period of time. Because P_f must be smaller than 50%, packets are more likely to be forwarded to the neighbors from the closer list, and the average direction of those neighbors points towards the receiver. Therefore, after overhearing enough packets, the adversary is able to figure out a direction along which he can search for the receiver.

To address the above weakness, we introduce an additional mechanism to smooth out the traffic trend by sending fake packets towards the opposite direction of the receiver. Fake packets are also used in the algorithm proposed in [2]. However, our work differs from theirs by that our goal of injecting fake packets is to minimize local information exposed to adversaries, whereas their goal is to modulate the whole traffic pattern in a network. The fake packet injection introduced here must be combined with the LPR routing protocol to take effect.

B. Fake Packet Injection

The basic idea of fake packet injection is that when a sensor node forwards a real data packet, it may generate a fake packet and transmit it to a neighbor randomly chosen from the $further\ list$. The transmission of the fake packet does not have to happen right after (or before) the real packet. A random delay can be introduced between them. Attracted by this fake packet, the adversary may trace to a wrong direction instead of the real next hop.

A system parameter M_{fake} specifies the maximum number of hops it will be forwarded away from the receiver. On one hand, a larger value for M_{fake} will lead to more traffic flowing away from the receiver, increasing the capability of misleading the adversary. On the other hand, a larger value for M_{fake} will also lead to higher energy consumption. It should be emphasized that M_{fake} has to be at least 2 hops. When M_{fake} is one hop, the next-hop sensor will not retransmit the fake packet, which can be detected by the adversary. When a node receives a fake packet, it does the following.

(1) The node decrements the TTL field (initialized to be M_{fake}) of the packet by one.

(2) If the TTL field is positive, the node randomly chooses a neighbor from its $further\ list$ and forwards the fake packet to that neighbor.

(3) If the TTL field is zero, the node discards the fake packet.

The injection of fake packets can effectively enhance the protection of the receiver's location privacy. However, the cost is also high. To control the tradeoff between energy consumption and protection strength, we introduce another system parameter P_{fake}, specifying the probability at which a node generates a fake packet when it forwards a real packet. The higher the value of P_{fake}, the more the number of fake packets that will be generated, and the more the energy that will be consumed. By adjusting this parameter, one can tune the tradeoff between security strength and energy cost. Based on the above design of fake packet injection, we can easily derive the following property.

Property 2: For any sensor i that is not in the immediate neighborhood of the receiver, under LPR with fake packet injection, a) the expected packet rate from i to *any closer neighbor* is the same, and b) the expected packet rate from i to *any further neighbor* is also the same.

Because no fake packets are sent to closer neighbors, directly from Property 1, we have that the expected rate from i to any closer neighbor is the same. Because fake packets are sent to further neighbors uniformly at random, we also have that the expected rate to any further neighbor is the same. However, the above two expected rates, one for closer neighbors and one for further neighbors, may be different, depending on the values of the system parameters.

Under LPR with fake packet injection, if the system parameters (P_f, M_{fake}, and P_{fake}) are appropriately set such that the following objectives are met, then it is very hard for an adversary to infer the direction towards the receiver based on packets overheard locally. A *further* (or *closer*) *direction* refers to a direction that moves *away* (or *closer to*) to the receiver.

- *Departure-rate objective*: At most sensors, outgoing packets are sent to all directions with nearly equal rates. Although the overall trend is that more real packets are sent to closer directions, this trend is balanced by fake packets sent to further directions.

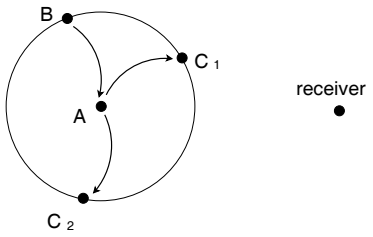


Fig. 2. Combine LPR with fake packets.

• *Arrival-rate objective*: At most sensors, arrival packets are coming from every direction with nearly equal rates. Although the overall trend is that more real packets come from further directions, this trend is balanced by fake packets coming from closer directions. An adversary cannot determine the direction of the receiver as he observes that packets come from all directions at uniform rates.

In the next section, we will analyze the impact of the system parameters on the performance of LPR with fake packet injection. The analytical results will answer the question of how to pick the values of the parameters in order to achieve the above objectives.

Next, we show that the problem described in Fig. 1 of Section III-C is unlikely to happen when LPR is applied together with fake packet injection. In Fig. 1, node A is the sensor where the adversary resides. Node B is the sensor that forwards a real packet to A. Nodes C_1 and C_2 are the neighbors to which A sends the real packet and the fake packet, respectively. They will forward the received packets, which reveal their locations to the adversary. We have explained that the adversary can identify the direction of the real packet through the relations among the transmissions made from B to A, from A to C_1, and from A to C_2. It means that fake packet injection has little effect when working with traditional routing protocols. However, when LPR is used, the direction to which the real packet is forwarded does not necessarily point towards the receiver. This implies that nodes B, A and C_1 are likely to not locate along a line. It is possible that the deviation of C_1 from the line of $B \rightarrow A$ is larger than that of C_2, as shown in Fig. 2. In this case, the adversary can hardly tell which of C_1 and C_2 is receiving the real packet. If he follows his strategy to choose the one with smaller deviation, he will trace to C_2, which is the wrong way.

In the above discussion, we assume that the packet transmitted from node B to node A is a real packet. If the packet itself is fake, which is possible when LPR is applied together with fake packet injection, then no matter where A forwards the packet, the adversary will trace to a wrong way.

V. Analysis

In this section, we analyze the properties of our defense scheme, i.e., LPR with fake packet injection.

Consider an arbitrary sensor i that is not in the proximity of the receiver. By Property 2, we know that the expected packet rate from i to *any closer neighbor* is the same, which is denoted as R_c. We also know that the expected packet rate from i to *any further neighbor* is also the same, which is denoted as R_f. There are three system parameters: i) P_f, the probability of forwarding a real packet to a further neighbor, ii) P_{fake}, the probability of generating a fake packet when forwarding a real packet, and iii) M_{fake}, the number of hops that a fake packet will be forwarded. First, we solve the problem of how to determine the appropriate values for the system parameters in order to meet the departure-rate objective, which requires i to send packets to all neighbors at equal expected rates, i.e., $R_c = R_f$. The arrival-rate objective for equalizing expected arrival rates will be discussed later.

In the following, we derive a formula that establishes the mathematical relation between R_c and R_f in the form of $R_f = F(P_f, P_{fake}, M_{fake})R_c$, where F is a certain function of the system parameters. Once F is identified, in order to achieve $R_c = R_f$, we can simply pick the values of the system parameters to satisfy $F(P_f, P_{fake}, M_{fake}) = 1$.

R_f has two components: $R_{f,1}$, the expected rate of *real packets* forwarded to a further neighbor due to randomized routing of LPR, and $R_{f,2}$, the expected rate of *fake packets* forwarded to a further neighbor.

$$R_f = R_{f,1} + R_{f,2} \tag{2}$$

By Property 1, we have

$$R_{f,1} = \frac{P_f \lambda}{1 - P_f} R_c \tag{3}$$

where $\lambda = \frac{n_c}{n_f}$, n_c is the number of closer neighbors for i, and n_f is the number of further neighbors. Next we establish the mathematical relation between $R_{f,2}$ and R_c. Let $R_{f,2}(t)$ be the expected rate of fake packets from i to a further neighbor *with the TTL field of the packets being* t.

$$R_{f,2} = \sum_{t=1}^{M_{fake}} R_{f,2}(t) \tag{4}$$

We assume that the local network/traffic conditions are roughly uniform in any sensor's neighborhood. Hence, the expected departure rates of fake packets, $R_{f,2}(t), t \in [1..M_{fake}]$, are about the same for i and its neighbors.

If two neighboring nodes have the same distance to the receiver, they will be in each other's further lists (Section IV-A). Let n_e be the number of i's neighbors that have the same distance to the receiver as i does. These neighbors are called *pseudo-further neighbors*. Let $\alpha = \frac{n_e}{n_f}$, which is the percentage of nodes in the further list that are pseudo-further neighbors. Note that i will receive fake packets from both its closer neighbors and pseudo-further neighbors.

For each $t \in [1..M_{fake})$, the expected rate of fake packets with TTL $= t$ sent from i to all further neighbors is equal to the expected rate of fake packets with TTL $= t + 1$ received by i from all closer or pseudo-further neighbors. The former is $n_f R_{f,2}(t)$, and the latter is $(n_c + n_e)R_{f,2}(t+1)$. Therefore, $n_f R_{f,2}(t) = (n_c + n_e)R_{f,2}(t+1)$, which can be rewritten as

$$R_{f,2}(t) = (\lambda + \alpha)R_{f,2}(t+1) \tag{5}$$

Recursively applying (5) to (4), we have

$$R_{f,2} = R_{f,2}(M_{fake}) \sum_{t=1}^{M_{fake}} (\lambda + \alpha)^{t-1} \tag{6}$$

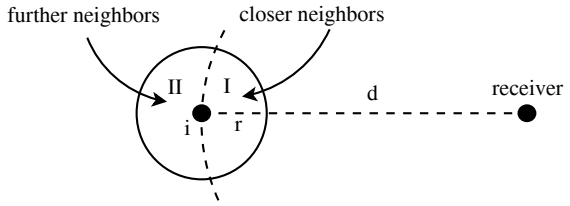


Fig. 3. Computing the value of λ, which is the ratio of the number of closer neighbors to the number of further neighbors. r is the transmission range of i, and d is the distance from i to the receiver.

We need to derive $R_{f,2}(M_{fake})$ as a function of R_c. The expected rate of real packets forwarded by i to all neighbors is $n_c R_c + n_f R_{f,1}$. The expected rate of fake packets with TTL $= M_{fake}$ sent by i to all further neighbors is $n_f R_{f,2}(M_{fake})$. For each real packet that i forwards, it generates a new fake packet (TTL $= M_{fake}$) with probability P_{fake}. Therefore, we have

$$(n_c R_c + n_f R_{f,1}) P_{fake} = n_f R_{f,2}(M_{fake}) \tag{7}$$

Applying (3), we have

$$R_{f,2}(M_{fake}) = \lambda(1 + \frac{P_f}{1-P_f}) P_{fake} R_c \tag{8}$$

Combining (6) and (8), we have

$$R_{f,2} = \lambda(1 + \frac{P_f}{1-P_f}) P_{fake} R_c \sum_{t=1}^{M_{fake}} (\lambda+\alpha)^{t-1} \tag{9}$$

By (2), (3), and (9), we have

$$\frac{R_f}{R_c} = \frac{P_f \lambda}{1-P_f} + \lambda(1 + \frac{P_f}{1-P_f}) P_{fake} \sum_{t=1}^{M_{fake}} (\lambda+\alpha)^{t-1} \tag{10}$$

If our objective is to ensure $R_c = R_f$, then the values of P_f, P_{fake} and M_{fake} should be selected to satisfy the following equation.

$$\frac{P_f \lambda}{1-P_f} + \lambda(1 + \frac{P_f}{1-P_f}) P_{fake} \sum_{t=1}^{M_{fake}} (\lambda+\alpha)^{t-1} = 1 \tag{11}$$

After picking the values for two parameters, the third one can be calculated.

The values of λ and α can be determined by counting the numbers of closer/further/pseudo-further neighbors. It may also be approximately calculated if we assume that the closer/further lists are constructed based on the neighbors' distances to the receiver and assume that the physical distribution of sensors is roughly uniform in a local neighborhood. In this case, λ is approximately equal to the ratio of the size of area I to the size of area II, and α is zero, as shown in Fig. 3. Let r and d be the transmission range of i and the distance from i to the receiver, respectively. We have

$$\lambda = \frac{r^2 \cos^{-1}\frac{r}{2d} + 2d^2 \sin^{-1}\frac{r}{2d} - r\sqrt{d^2 - \frac{r^2}{4}}}{\pi r^2 - (r^2 \cos^{-1}\frac{r}{2d} + 2d^2 \sin^{-1}\frac{r}{2d} - r\sqrt{d^2 - \frac{r^2}{4}})} \tag{12}$$

For each sensor, because λ is different, the parameter values calculated based on (11) will also be different. However, for all sensors distant from the receiver, the size of area I is close to the size of area II, and λ can be approximated as one. For these sensors, (11) can be simplified as

$$\frac{P_f}{1-P_f} + (1 + \frac{P_f}{1-P_f}) P_{fake} M_{fake} = 1 \tag{13}$$

Therefore, as an approximation, the parameter values can be set the same for all distant sensors based on (13). If the sensor network covers a large area and most sensors are distant from the receiver, then we may practically pre-set the parameter values for all sensors based on this equation.

Now we show that, if the departure-rate objective is met, i.e., $R_c = R_f$, then the arrival-rate objective will also be met, which requires sensor i to receive packets from all neighbors at equal expected rates. Recall our assumption that the local network/traffic conditions are roughly uniform in a sensor's neighborhood. It means that the neighbors of i send packets to their neighbors at equal expected rates and, furthermore, those expected rates are equal to the departure rates from i. Consequently, i receives packets from the neighbors at equal expected rates.

Finally, we relax the arrival-rate/departure-rate objectives and generalize the formula for setting the system parameters in order to make tradeoff between security strength and overhead. The receiver's location privacy is best protected when $\frac{R_f}{R_c} = 1$ because, by following overheard packets, an adversary will move randomly in the network. However, as we will see shortly, the communication overhead of our defense scheme is a function of $\frac{R_f}{R_c}$. The higher the value of $\frac{R_f}{R_c}$ is, the higher the overhead will be. One way to control the overhead is to relax the arrival-rate/departure-rate objectives by allowing $\frac{R_f}{R_c} < 1$. Reducing the overhead can be achieved by lowering $\frac{R_f}{R_c}$. However, with R_c greater than R_f, when an adversary follows the overheard packets, despite being frequently diverted away, he will make statistical progress towards the receiver over the long run. The lower the value of $\frac{R_f}{R_c}$, the higher the speed of the adversary moving towards the receiver, and the weaker the security for the receiver. We characterize the overhead ratio (denoted as OH) as the expected rate from i to all further neighbors divided by the expected rate from i to all closer neighbors.

$$OH = \frac{n_f R_f}{n_c R_c} = \frac{1}{\lambda}\frac{R_f}{R_c} \tag{14}$$

If a sensor-network application specifies the maximum allowed overhead ratio, we can calculate the maximum value for $\frac{R_f}{R_c}$ and then determine the values of the system parameters based on (10) or the following approximation formula with $\lambda = 1$ for distant sensors,

$$\frac{P_f}{1-P_f} + (1 + \frac{P_f}{1-P_f}) P_{fake} M_{fake} = \frac{R_f}{R_c} \tag{15}$$

which is the generalization of (13).

Our defense system can be further generalized by allowing the expected rate R_f to vary among further neighbors and R_c to vary among closer neighbors. This generalization can be realized as follows: When a closer (or further) neighbor is selected as the next hop for a packet, instead of choosing it from the closer (further) list uniformly at random, we may choose it based on a certain non-uniform probability distribution on the list. For example, while keeping the

overhead ratio smaller than one, we may forward more fake packets to a selected set of further neighbors such that the expected rates to them are as high as the expected rates to the closer neighbors, whereas the expected rates to other further neighbors are smaller. How to use such a generalization to enhance the receiver's security is a subject of our future work.

In the next section, we will use simulations to show how different values of the system parameters affect the performance and overhead of our defense scheme. We will compare our scheme with others and demonstrate how the tradeoff between performance and overhead can be tuned through the system parameters.

VI. Performance Evaluation

We evaluate the performance of our new methods through simulations based on three criteria: delivery time, strength of privacy protection, and energy cost, which will be defined shortly. We compare our methods with single-path routing and two other location-privacy protection schemes: Phantom routing in [1] and DEFP in [2]. Single-path routing is used as the baseline scheme. Although Phantom routing is originally designed for protecting the location privacy of source nodes, to some extent it can also be used to protect the receiver's location privacy. We assign the random walk distance in the directed random walk phase of Phantom routing to be 10 hops. For DEFP, we use the default configuration settings in the original paper [2]. For LPR, the further/closer lists are calculated based on the Euclidean distances from the nodes to the receiver. When evaluating the strength of privacy protection, we first study the scenario where fake packets are not generated and then move to the scenario where fake packets are used. We will see that, with the significant energy overhead, fake packet injection is able to enhance the protection strength by two orders of magnitude or more.

A. Delivery Time

Delivery time is the time it takes a packet to move from its source node to the receiver under a certain routing protocol. In our simulations, it is measured as the average number of hops that packets from a selected source node traverse before reaching the receiver. The baseline single-path routing scheme has the smallest delivery time because the packets always follow the shortest path to the receiver. For other schemes, the packets may follow longer paths due to randomization introduced in the routing process.

The impact on the routing path length by LPR and other schemes is examined in Fig. 4. Our simulation uses a sensor network of 2,500 nodes with the average number of neighbors being 8. In the figure, $baseline$ represents the single-path routing scheme, and LPR is assigned with different P_f values: 0%, 12.5%, 25.0%, and 37.5%. Fake packet injection is turned off in this simulation.

Comparing with single-path routing, which is optimal in delivery time, all other schemes have longer routing paths. The average path length in LPR increases when the value of P_f increases. When P_f becomes 37.5%, the average path length becomes 10 times of the length of the shortest path. This suggests that P_f should not be assigned a large value unless extremely strong protection for location privacy is required. In fact, from the simulation results in the next subsection, we will see that strong protection can be achieved with small P_f values when fake packets are used.

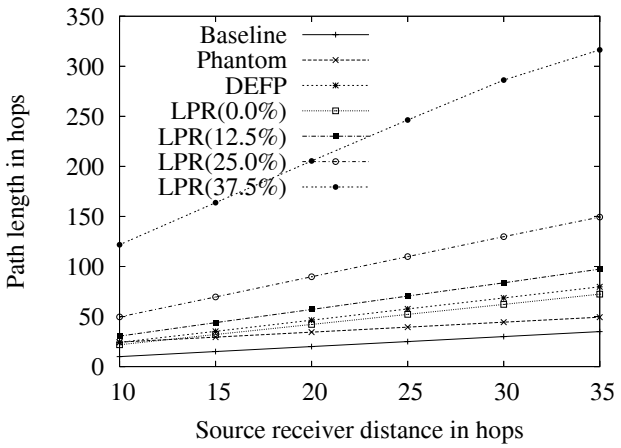


Fig. 4. Average path length in number of hops. LPR is assigned four different P_f values: 0%, 12.5%, 25.0%, and 37.5%. $Baseline$ represents the baseline single-path routing scheme.

B. Strength of Privacy Protection

To evaluate the strength of location-privacy protection, we use two criteria: the safe time of the receiver and the attack time of the adversary. The receiver's *safe time* is measured as the number of packets delivered before the receiver is captured. The adversary's *attack time* is measured as the number of moving steps (from one sensor location to a neighbor) the adversary has to make before he reaches the receiver. We perform the simulations on a sensor network of 900 nodes, among which 100 randomly-selected ones are source nodes that periodically generate data packets for the receiver. The *source period* is defined as the time between two packets generated from a source node. The initial distance from the adversary to the receiver is 15 hops. The adversary has the characteristics described in Section III-B. In his attack strategy, the adversary stays in one position to overhear packets, and move to the next location based on the policy described in Section III-C. If no packet is overheard for a duration of 100 source periods, the adversary backtracks to his previous location. The adversary remembers 5 steps in his moving history. After his history record is exhausted, he walks randomly until catching a packet. We set a time limit for our simulation, which is the time for 500,000 packets to be delivered. The program terminates if the adversary cannot capture the receiver in the time limit.

We first study the case without fake packets. The receiver's location privacy is protected only through path randomization. Fig. 5 shows the simulation results on the safe time provided by single-path routing (baseline), Phantom routing, DEFP, and LPR with varying P_f values. When single-path routing is used, the receiver is captured after about 600 packets are delivered. Phantom routing improves the safe time to around 890 packets. LPR provides different safe times when different P_f values are used. When P_f is greater than 30%, the safe time of LPR is

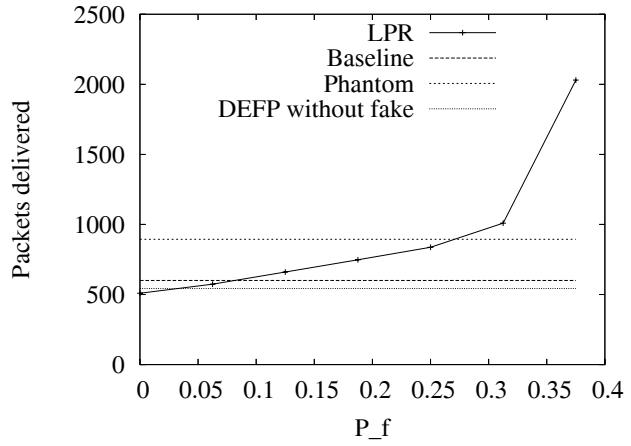


Fig. 5. Safe time provided by different schemes without fake packet injection. Safe time is measured in the number of packets delivered before the receiver is captured. LPR is assigned with varying P_f value.

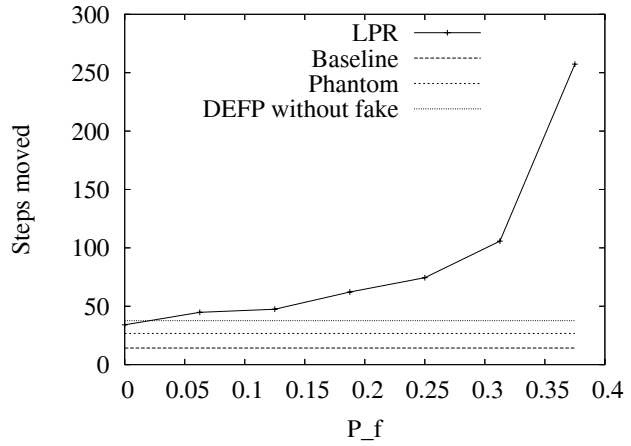


Fig. 6. Attack time under different schemes without fake packet injection. LPR is assigned with varying P_f value.

several times longer than those of other schemes, but it also has longer delivery time and higher energy cost. Without fake packets, the safe time of DEFP is even lower than single-path routing (baseline).

An interesting observation is that the safe time of LPR is slightly worse than that of single-path routing when P_f is very small. That can be explained as follows. First, a very small value for P_f leads to insignificant randomization in routing paths, and the effect of misleading the adversary is negligible. Second, the path randomization allows the packets to better utilize the network bandwidth, which has a greater impact than the slight increase in average path length when P_f is small.

Fig. 6 presents the simulation results on the attack time, i.e., the number of moving steps the adversary has to take before he reaches the receiver. The figure shows that LPR makes the adversary to move more steps than other schemes, even when P_f is as small as 0.025. When P_f is as large as 0.35, the number of steps that adversary has to move in LPR is many times of that in other schemes.

The above simulations do not use fake packets. Although routing randomization achieves a certain level of protection, the security it provides for the receiver is insufficient for critical applications. The safe time of the receiver and the attack time of the adversary under LPR are in the same order as or only one order of magnitude higher than those under the baseline single-path routing scheme. The reason is that, although routing paths are randomized, the overall traffic trend remains flowing towards the receiver. Consequently, by tracing the packets, the adversary will statistically make progress towards the receiver. Now, if LPR is combined with fake packets, the adversary will overhear packets flowing from and to all directions at much more uniform rates, which makes it hard for the adversary to deduce correct information from locally transmitted packets. Table I shows the performance of single-path routing (baseline), Phantom routing, DEFP with fake packets, and LPR with fake packets. Note that the first two schemes do not use fake packets by design. In this simulation, the parameters in LPR are set as follows: $P_f = 0$, $M_{fake} = 7$, and $P_{fake} = 44\%$. For each hop a real packet moves, the average number of hops taken by the corresponding fake packet is $P_{fake} \cdot M_{fake} = 3.1$. We assign the parameters with the above values because the resulting energy cost is approximately equivalent to the energy cost of DEFP, which is about 7 times the energy cost of the baseline single-path routing scheme. It is worth noting that such a high energy cost is not always required. The next subsection shows the performance of LPR in varied energy cost levels. In Table I, the safe time and the attack time are defined as before. The path length is the average number of hops on the routing paths from the source nodes to the receiver. This value indicates the delivery time under a given routing scheme. The table shows that DEFP can only improve safety moderately, which means DEFP is not a good solution for the packet-tracing attack. However, LPR with fake packet injection improves safety significantly. Its save time is more than 100 times higher than DEFP, and its attack time is 10 times that of DEFP.

TABLE I
SIMULATION RESULTS FOR LPR AND DEFP WITH FAKE PACKETS

	Safe time	Attack time	Path length
Baseline	600.5	15.17	15.07
Phantom	894.3	26.74	26.77
DEFP+fake	3030.5	185.1	31.27
LPR+fake	370299	1875.7	27.21

C. Tradeoff Between Energy Cost and Protection Strength

For those applications where the energy cost is as important as the safety of the receiver, one may achieve a balance between energy cost and protection strength by adjusting the value of the system parameter P_{fake}. We have simulated LPR with fake packet injection for different P_{fake} values to exam the influence of P_{fake} on the performance/overhead tradeoff. Other parameters remain the same as previous. The energy cost is measured as the ratio of *the total number of hops that all packets (real or fake) are forwarded under a given scheme* to *the total number of hops forwarded under the baseline single-path routing scheme*. Fig. 7 and 8 present the simulation results on safe time and energy cost, respectively.

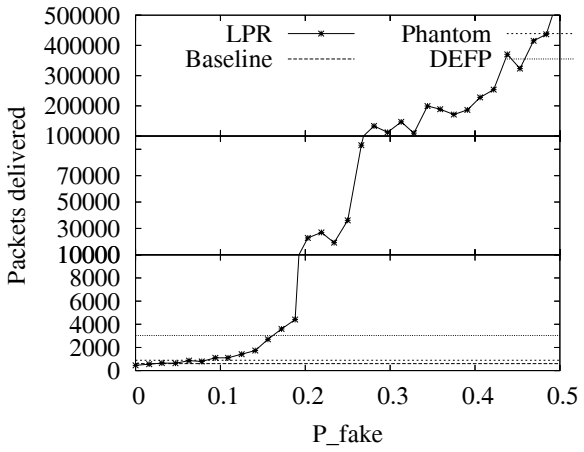


Fig. 7. Safe time provided by LPR with varying P_{fake} value. LPR is combined with fake packet injection. M_{fake} = 7, P_f = 0.

The Y axis of Fig. 7 is not linearly scaled so that the data can be more clearly presented. We find that the safe time of LPR grows linearly when P_{fake} increases from 0 to 14%. After that, it climbs drastically faster. When P_{fake} reaches 50%, the safe time becomes one thousand times as long as that of the baseline scheme. At the mean time, the energy cost is always linear to the value of P_{fake}. We are interested in the region of P_{fake} between 15% and 30%, where the energy cost is acceptable while excellent protection strength is achieved. Fig. 9 demonstrates the tradeoff between safe time and energy consumption and compares LPR and DEFP. The curves are obtained by varying the probability of fake packet generation (P_{fake}). We can see that, with energy increased, the safe time archived by LPR climbs much more quickly than that under DEFP.

VII. Conclusions

In this paper, we design LPR, a location-privacy routing protocol, and combine it with fake packet injection to protect the location privacy of the receiver in a sensor network. We study the packet-tracing attack, in which an adversary traces the location of a receiver by eavesdropping and following the packets transmitted in the sensor network. This attack cannot be effectively countered by the existing approaches. Our system addresses the attack in two ways. First, LPR randomizes the routing paths. Second, fake packet injection attempts to make both incoming packets and outgoing packets uniformly distributed in all directions at a node. This makes it very hard for an adversary to infer the location of or the direction to the receiver. Moreover, the tradeoff between protection strength and energy consumption is made tunable through two system parameters. We perform extensive simulations to evaluate LPR with false packet injection based on three criteria: delivery time, protection strength, and energy cost. The results show that, comparing with other methods, LPR with fake packet injection provides stronger protection for the receiver's location privacy. In the future work, we will extend our study to networks with multiple receivers, and we will also formally analyze the performance of our scheme.

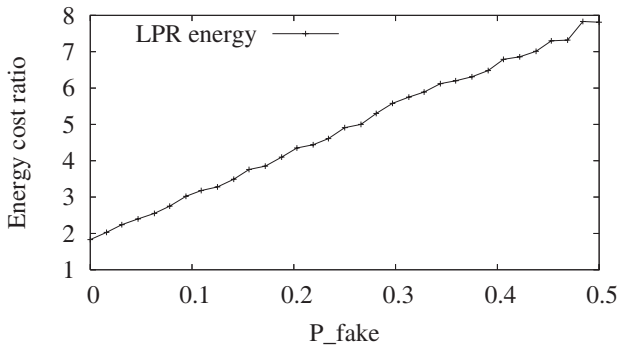


Fig. 8. Energy cost of LPR with varying P_{fake} value. LPR is combined with fake packet injection. M_{fake} = 7, P_f = 0. Energy cost is measured as a ratio to baseline sing-path routing.

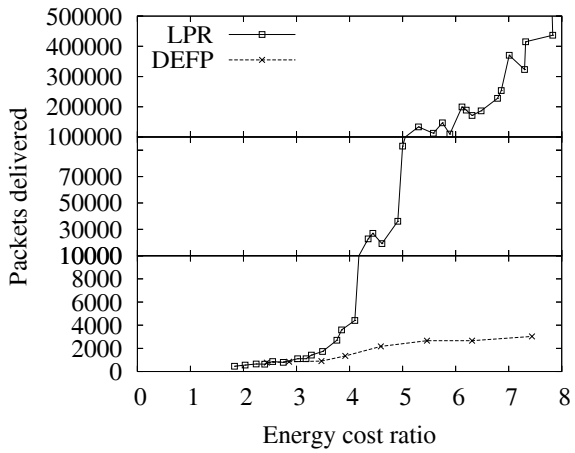


Fig. 9. Tradeoff between safe time and energy consumption.

References

[1] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *Proc. 25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2005.

[2] J. Deng, R. Han, and S. Mishra, "Countermeasures against traffic analysis attacks in wireless sensor networks," in *Proc. CerateNet Conference on Security and Privacy in Communication Networks (SecureComm)*, 2005.

[3] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: mobile networking for smart dust" in *Proc. 5th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 1999.

[4] F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, "Wireless sensor networks: a survey, computer networks," *Computer Networks*, vol. 38, pp. 393-422, 2002.

[5] A. Perrig, R. Szewczyk, D. Tygar, V. Wen, and D. Culler, "Spins: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521-534, 2002.

[6] L. Eschenaur and V. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. 9th ACM Conference on Computer and Communications Security*, 2002.

[7] M. Spreitzer and M. Theimer, "Providing location information in a ubiquitous computing environment," in *Proc. 14th ACM Symposium on Operating System Principles*, 1993.

[8] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion," in *Proc. IEEE/CreateNet International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm)*, 2005.

[9] M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald, "Privacy-aware location sensor networks," in *Proc. 9th USENIX Workshop on Hot Topics in Operating Systems (HotOS IX)*, 2003.

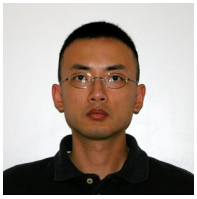
[10] J. Al-Muhtadi, R. Campbell, A. Kapadia, M. D. Mickunas, and S. Yi, "Routing through the mist: privacy preserving communication in ubiq-

uitous computing environments," in *Proc. 22nd IEEE International Conference of Distributed Computing Systems (ICDCS)*, 2002.
[11] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous connections and onion routing," *IEEE J. Select. Areas Commun.*, vol. 16, no. 4, pp. 482-494, May 1998.
[12] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Mask: anonymous on-demand routing in mobile ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 5, no. 9, pp. 2376-2385, Sept. 2006.
[13] J. Deng, R. Han, and S. Mishra, "Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks," in *Proc. IEEE International Conference on Dependable Systems and Networks(DSN)*, 2004.
[14] C. Ozturk, Y. Zhang, and W. Trappe, "Source-location privacy in energy-constrained sensor network routing," in *Proc. 2004 ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN) in conjunction with ACM Conference on Computer and Communications Security*, 2004.
[15] Y. Xi, L. Schwiebert, and W. Shi, "Preserving source location privacy in monitoring-based wireless sensor networks," in *Proc. 2nd International Workshop on Security in Systems and Networks (SSN)*, in conjnuction with IPDPS, 2006.
[16] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," in *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, Feb. 1999.
[17] Z. Haas, J. Halpern, and L. Li, "Gossip based ad hoc routing," in *Proc. IEEE INFOCOM*, 2002.
[18] C. Barrett, S. Eidenbenz, and L. Kroc, "Parametric probabilistic sensor network routing," in *Proc. 2nd ACM international Conference on Wireless Sensor Networks and Applications*, 2003.
[19] H. Lim and C.Kim, "Flooding in wireless ad hoc networks," *Computer Commun. J.*, vol. 24(3-4), pp. 353-363, 2001.
[20] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proc. 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2000.

Ying Jian received the B.E. degree in 2001 and the M.E. degree in 2004, both in computer science, from Tsinghua University, China. He is currently a Ph.D. candidate at the Department of Computer and Information Science and Engineering, University of Florida. His research interests include QoS and security in multihop wireless networks and sensor networks.

Shigang Chen received his B.S. degree in computer science from University of Science and Technology of China in 1993. He received M.S. and Ph.D. degrees in computer science from University of Illinois at Urbana-Champaign in 1996 and 1999, respectively. After graduation, he had worked with Cisco Systems for three years before joining University of Florida in 2002. His research interests include network security and wireless networks. He received IEEE Communications Society Best Tutorial Paper Award in 1999 and NSF CAREER Award in 2007. He was a guest editor for ACM/BALTZER JOURNAL OF WIRELESS NETWORKS (WINET) and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGIES. He served as a TPC co-chair for the Computer and Network Security Symposium of IEEE IWCCC 2006, a vice TPC chair for IEEE MASS 2005, a vice general chair for QShine 2005, a TPC co-chair for QShine 2004, and a TPC member for many conferences including IEEE ICNP, IEEE INFOCOM, IEEE ICC, IEEE Globecom, etc.

Zhan Zhang is a software engineer in Cisco Systems. He received his M.S. degree in computer science from Fudan University of China in 2003, and Ph.D in Computer and Information Science and Engineering from University of Florida in 2007. His research interests include overlay networks, wireless sensor network and network security.



Liang Zhang is a Ph.D. candidate in the Computer and Information Science and Engineering Department at the University of Florida. He received B.E. and M.E. degrees in computer science from Tsinghua University of China in 1999 and 2001, respectively. After that, he had worked in Oracle R&D Center in China for one year. He started his PhD program in 2003. His current research interests include multihop wireless networks, sensor networks and network security.