# An Anti-Tracking Source-Location Privacy Protection Protocol in WSNs Based on Path Extension

Wei Tan, Ke Xu, *Senior Member, IEEE*, and Dan Wang, *Senior Member, IEEE*

***Abstract*—In the application field using sensor networks to monitor valuable asset, source-location anonymity is a serious concern. As a series of event packets are reported to the base station, adversaries eavesdropping on the network can backtrack to the source through traffic analysis and the RF localization techniques. This leakage of contextual information will expose sensitive or precious objects and bring down the effectiveness of sensor networks. Existing techniques such as phantom routing or source simulation are proposed to discourage the adversaries, both of which trade energy for security. In this paper, we propose a new scheme, called path extension method (PEM), providing strong protection for source-location privacy. It performs quite well even though an object occurs near the base station, while other methods cannot protect the source well in this case. In PEM, fake sources are generated dynamically after the source sends event messages to the base station, which makes it much more flexible. Fake sources form several fake paths in the network and an adversary will be induced farther away from the source if it is entrapped by any of them. The theoretical and simulation results show that PEM is efficient in protecting source-location privacy with minimal message delivery delay and acceptable overhead.**

***Index Terms*—Environment monitoring, source-location privacy, traffic analysis, wireless sensor networks (WSNs).**

## I. Introduction

COMPARED with traditional networks, wireless sensor network (WSN) is a new network structure [1]. Owing to the low price and easy deployment, it has been used in various fields such as military applications and study of endangered species. A sensor network typically includes lots of sensor nodes and several base stations (also known as sinks). The sensor nodes are responsible for environment monitoring and sending event-reporting packages to the base stations. They are usually power limited and poor in data processing with small storage space. However, the form factor (smaller size) helps them to be inconspicuous in the sensing area and therefore widely used. In contrast, the base stations in a sensor network are more powerful with large storage, strong data processing capability, and greater bandwidth. All of the data sensed by sensor nodes are collected and processed by the base stations. Users can access the data via the Internet, so the base stations are considered to be gateways of the sensor network and the Internet. When an object occurs at someplace in the monitoring area, the sensor nodes close to it turn to be sources. They will generate event-reporting packets with the gathered sensing data and send them to the base stations periodically through multihop wireless transmission. As mentioned above, sensor nodes are usually resource constrained, and their computing and storage capacity as well as the RF communication range are very limited. Moreover, the sensor nodes are powered by batteries and it is difficult or even impossible to replace the batteries for them. Due to these reasons, sensor energy is especially valuable, which requires routing techniques of sensor networks to be designed with meticulous care, and energy saving becomes a primary objective.

The unreliability of wireless links and the broadcast property of physical layer provide potential for an adversary to overhear the network [2]. In some sensor network applications such as monitoring of endangered species or real-time battlefield environment, source-location privacy protection is an important concern in routing protocol design. Source-location leakage exposes the locations of monitored objects (endangered species or soldiers) and the relevant information will be utilized by malicious attackers (e.g., hunters). This will result in a serious hidden danger since an adversary may start eavesdropping at a base station. As soon as an event packet is sent to the base station, the attacker can overhear it and then determine the message's direct sender through the RF positioning technology. It moves to the sender and then continues to listen until it reaches the source hop-by-hop.

Hereby, we propose a new scheme, named path extension method (PEM), to provide strong protection for source-location privacy. It performs quite well even though an object occurs near the base station while other methods have poor performance in this case. The theoretical and simulation results show that PEM is efficient in protecting source privacy with minimal delivery delay and acceptable overhead. PEM is essentially a method using fake sources, but it has much more flexibility than other schemes. Instead of designating fixed fake sources after network deployment, fake sources are produced dynamically in PEM after the source sends event messages to

Manuscript received January 13, 2014; revised June 21, 2014 and July 30, 2014; accepted August 02, 2014. Date of publication August 12, 2014; date of current version October 21, 2014. This work was supported in part by the National Science and Technology Major Project (2012ZX03005001), the NSFC Project (61170292, 61161140454), the 973 Project of China (2012CB315803), and the 863 Project of China (2013AA013302).

W. Tan and K. Xu are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: tan-w12@mails.tsinghua.edu.cn; xuke@tsinghua.edu.cn).

D. Wang is with the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: csdwang@comp.polyu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JIOT.2014.2346813

the base station. A series of fake sources form several fake paths and an adversary will be induced farther away from the source if it is entrapped by any of them, which significantly prolongs the safety period.

This paper is organized as follows. Section II introduces related work. Section III describes the network and threat models. Section IV presents our PEM scheme. Section V analyzes the privacy properties of PEM. Section VI describes simulation experiments and performance evaluation of PEM. Finally, Section VII concludes this paper.

## II. Related Work

There has been a lot of work examining the contextual privacy in conventional networks. Some provide destination privacy [3], [4], while the others protect source privacy. However, many of the solutions developed for general networks [5], [6] are inappropriate for sensor networks due to the limited capabilities of nodes and most protocols designed for sensor networks are optimized for the same reason. Karlof and Wagner proposed threat models and security goals for secure routing in WSNs [7]. They analyzed the security property for all of the major routing protocols deployed in sensor networks. Moreover, Alomair *et al*. presented a statical framework for modeling, analyzing, and evaluating anonymity in sensor networks [8].

Since Ozturk *et al*. puts forward the problem [9], source-location privacy in sensor networks has been carefully studied [10], [11]. A panda-hunter game is introduced in [9] to formalize this problem. The hunter tries to backtrack the routing path of messages and eventually capture the panda. Solutions including fake messaging and phantom routing technique are devised in [12] to enhance the source-location privacy. These two methods can be combined with basic routing techniques like flooding or single-path routing. More discussions in [12] are given to the technique of phantom routing. It consists of two phases: one message is first transferred to an arbitrary node named phantom source through a random walk and then delivered to the base station through a subsequent flooding/single-path routing stage. It increases the difficulty for an adversary to track the source since messages come to the base station through different paths. According to [12], phantom single-path routing is more efficient than phantom flooding in terms of either safety period or energy consumption. Another algorithm using random walk is proposed in [13]. Li and Ren also proposed a two-phase routing process by transmitting messages to randomly selected intermediate node(s) before they are transmitted to the base station [10].

To withstand attacks under a global eavesdropper model, Mehta *et al*. proposed two techniques in [14], which are periodic collection and source simulation. The periodic collection method which has all of the nodes sends packets synchronously and periodically whether they have real data to send or not, which makes the global eavesdropper unable to determine the source from network traffic patterns. This mechanism is not suitable for real-time applications or sensor networks with high data-sending rate due to its high latency and large overhead. In the source simulation technique, a tradeoff between privacy and overhead has been made in contrast to periodic collection method. Multiple candidate traces are created to simulate the traffic generated by the source whose movements are taken into account. For the sake of reducing overhead of dummy traffic generation, Shao *et al*. developed a scheme called FitProbRate [15] based on the work in [16]. In this scheme, fake packets are not injected into the network at a constant rate, but following a probabilistic distribution which significantly improves the performance of fake messaging.

Two schemes are presented by Wang *et al*. in [17]. One is random parallel routing and the other is weighted random stride routing. The former routes every message on a randomly selected path from the source to the base station. It is not satisfactory because messages from any candidate path will pull the adversary toward the source. The later method divides neighboring nodes of the current sensor into different strides and weights them according to the possible length of routing path when a message chooses to go through that part. Both methods exploit randomness to obtain path diversity as random walk proposed in [12]. Pongaliur and Xiao presented a solution called SPENA based on one-way hash chain in [18]. A packet reconstruction model against eavesdropping and node comprise attack is also included.

In this paper, we devise a new scheme called PEM, which employs the technique of fake messaging to provide stronger source-location privacy protection. In [12], a *Persistent Fake Source* routing strategy is proposed, where fake sources are chosen by the source in a distributed manner and the locations of the persistent fake sources are fixed. It misses the best opportunities to mislead adversaries and the fake sources with unchanged locations are easy to identify for adversaries. A new concept of cyclic entrapment method (CEM) is proposed in [19], which is essentially a method using fake sources as well. In CEM, loops are configured in advance after deployment of the sensor network and every loop consists of an ordered sequence of nodes. Traffic is generated in a loop when it overlaps with the routing path of event-reporting messages. Even though loops in CEM inject fake messages immediately on real messages from the source going through them, an adversary will eventually discover the trap and return to the routing path of event-reporting messages. Moreover, the performance of CEM closely relies on the quantity and length of loops deployed in the sensor network.

PEM overcomes the disadvantages of the above methods. In PEM, fake sources are produced dynamically and form fake paths as soon as the source sends event-reporting messages to the base station. The locations and number of fake sources are ever changing. Once an adversary is induced in fake paths, it will be led farther away from the source until the end of the fake paths. We also adopt the ideal in [15], making fake sources in PEM inject messages at different rates to enhance its ability of source privacy protection and save precious energy at the same time. The theoretical and simulation results show that PEM is more efficient in protecting source privacy with minimal delivery delay and acceptable overhead. It performs quite well even though an object occurs near the base station while other methods have poor performance in this case.

## III. Network and Threat Model

The problem formalization is described in this section. We declare assumptions about the monitoring sensor networks and attack strategies that may be used by adversaries.

### A. Network Model

*1) Secure Base Station:* In a monitoring sensor network, a lot of sensor nodes are *homogeneously distributed* in a large area. Several base stations may disperse over the whole network to collect and process the sensing data from the lightweight sensor nodes. The base stations are secure with much greater computational capability, so it is assumed to be impossible to filch any information from them. One of the goals of deploying the base stations is to guarantee that every sensor node can deliver its data to at least one base station not far away. Without loss of generality, in this paper, only one base station is assumed to exist in the sensor network.

*2) Sending Encrypted Event Messages Periodically:* The sensor nodes in a monitoring WSN keep their eyes open all the time. When an object appears at somewhere in the environment, sensors near it will realize the case and start to collect information about the object. Then event packets are generated and sent to the base station periodically. Due to the bandwidth limitation of sensor nodes, a series of messages would be sent to the base station while the object is present. In military application, real-time messages are requested to report back to the headquarters, which makes it a matter of course to send messages continuously. Contextual information such as sensor ID and time stamp is carried in the messages so that the base station can know where and when the monitored object is present. Encryption technique is employed to ensure that an adversary will know nothing about the content of event messages.

*3) Single Source:* Multiple sensor nodes may detect the occurrence of object and every node will send messages to the base station. However, this causes a lot of data redundancy and consumes too much energy of the network. Besides, the more messages are reported back to the base station, the easier an adversary can trace back to the source. Based on this consideration, we assume that there is only one source in the network to report an event all the time. When the object moves to a new location, it triggers another sensor to be the source.

### B. Threat Model

*1) Backtracking Through Eavesdropping:* A sensor node can communicate with its neighbors in the range of radio. Whenever a sensor receives a packet, it broadcasts the message or simply discards it. Adversaries can eavesdrop on communications between sensors. It is assumed that an adversary has the same eavesdropping range as the radio communication range of sensor nodes [1]. Although an adversary cannot obtain the exact content of the messages intercepted, the direct sender of the messages can be determined using traffic analysis or RF localization techniques. Adversaries overhear at the base station at the very beginning. When intercepting a message, it moves to the location where the message came from. Then, it eavesdrops on the communications between the current node and its neighboring nodes until backtracking to the source hop-by-hop.

**Algorithm 1.** Strategy used by sophisticated adversaries

```
1: An adversary moves from node A to node B after it overhears a message;
2: The adversary starts the timer and sets the timeout interval to be δ′;
3: while (keep listening at node B) do
4:   if (overhear a message) then
5:     determine which node it is from, say C;
6:     if (C=A or C=B or historicalLocations.find(C)) then
7:       drop the message;
8:     else
9:       historicalLocations.push(C);
10:      move to node C;
11:      break;
12:    end if
13:  else if (timer timeout) then
14:    node X = historicalLocations.top();
15:    historicalLocations.pop();
16:    roll back to node X;
17:    break;
18:  else
19:    do nothing;
20:  end if
21: end while
```

*2) Historical Locations Recording and Back Rolling:* In general, adversaries have much larger storage than sensor nodes. So, we assume that they would record every location they have been to. Only in this way can the adversary avoid getting into circulations generated by fake sources that probably exist in the network. Circulations make it difficult for adversaries to track the source. In order to cope with this situation, a sophisticated adversary will check the historical locations after determining where a message comes from. Only if the message comes from a completely new sensor, the adversary would move to that node. Otherwise, it would ignore the message and keep listening at the current location. It is possible that a patient adversary cannot overhear anything for a long time. In this case, the adversary may roll back to the latest one among the recorded locations. Then this location will be removed from the record of historical locations. As depicted in Fig. 1, if an adversary traces back to sensor $S_4$ and then receives no message for some time $\delta'$, it will roll back to sensor $S_3$ and remove $S_4$ from the historical location record. The adversary returns to $S_1$ and eventually receives messages from another route. When it traces back to $S_5$, it intercepts a message from $S_4$ again and moves to it because $S_4$ is not in the historical locations any more. The tracking strategy used by adversaries is summarized in Algorithm 1.

### C. Local Eavesdroppers Versus Global Eavesdroppers

Local eavesdroppers: Adversaries who have only a local view of network traffic. Global eavesdroppers: Adversaries who

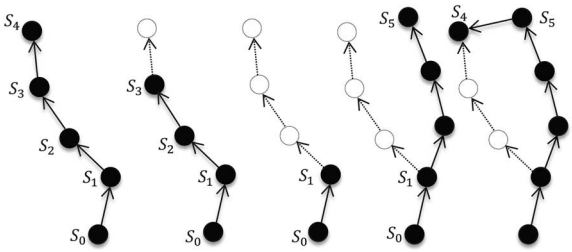



Fig. 1. Adversaries rolling back to historical locations.

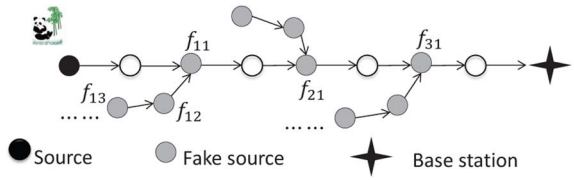



Fig. 2. PEM scheme.

eavesdrop on the entire network and have a complete view of network traffic. A global eavesdropper can easily infer the locations of monitored objects because the sensor nodes that initiate communication with the base station are usually close to the objects.

Some research works focus on privacy-preserving communication methods in the presence of a global eavesdropper. They assume that the adversary deploys his own set of sensor nodes to monitor the communication in the target network and thus gets a global view of network traffic. The technique of periodic collection [14], [20] is near-perfect to withstand attacks under a global eavesdropper because all of the nodes send packets synchronously and periodically whether they have real data to send or not, which makes a global eavesdropper unable to distinguish the real source. However, it consumes a lot of precious energy from sensor notes and incurs high latency because a specified amount of messages are allowed to be sent in a period instead of sending all of the messages continuously and immediately after their generations. Source simulation [14], [20] is a simple improvement of periodic collection. In the periodic collection method, every sensor node is a potential source node. Instead, in the source simulation approach, only a set of virtual sources (much smaller than the size of the network) are selected and simulated in the field. Since the virtual sources keep on generating traffics after network deployment, the overhead is still unaffordable. Li *et al*. proposed a multiintermediate nodes scheme to provide global source-location privacy [10], in which the randomly selected intermediate nodes are similar to the virtual sources described above. In [21], a small number of stealthy permeability tunnels are used to scatter and hide the communication patterns against a global eavesdropper. Specifically, wormhole nodes are deployed for preserving the source-location privacy and mobile ferry stations are used to hide the base station. In this approach, the randomly deployed wormhole nodes have higher priorities to relay messages, which makes a sophisticated global eavesdropper easily deduce the locations of them by performing traffic analysis.

Sensor networks can support a wide range of applications, and different applications may have different requirements. The presence of global eavesdroppers is usually a strong assumption and difficult to achieve, especially in a large monitoring area (such as $5000$ m $\times$ $5000$ m in this paper), so most of the privacy-preserving routing techniques developed for sensor networks focus on the assumption of local eavesdroppers, including *random walks* [7], [12], [17], and *cyclic entrapment* [19]. It is usually harder to protect source-location privacy under a global eavesdropper and the corresponding solutions are more costly with respect to energy and latency [22]. In this paper, we take the conception of local eavesdropper as well and provide an alternative for making deliberate choice to balance between source-location privacy and energy consumption in practical applications.

## IV. PEM

In the method using fake sources, the first issue to be solved is how to create fake sources and there exist two approaches in the literatures [12], [17]. One is generating fake sources after network deployment and activating them when they receive messages from the source. In order to save precious energy of WSNs, it is better to let fake sources send messages only after an event is observed. The other is to select fake sources by the real source on the basis of their distance to the sink. No matter which method is used, fake sources are static after they are designated. Therefore, they will be identified by adversaries sooner or later and the adversaries would record the locations of them. Then an adversary can return to the sink and choose a right direction toward the real source. In PEM, a dynamic fake source generating strategy is developed to entrap adversaries and strengthen the privacy protection level of the real source.

### A. Generation of Initial Fake Sources

In our proposed scheme, PEM, several fake sources are produced when a real source starts sending messages to the base station. As illustrated in Fig. 2, the first batch of fake sources is selected from nodes on the path between the real source and the base station, which are called *initial fake sources*. When a monitored object, say a panda, occurs in the network, the sensor node nearest to it turns to be a source. It then sends event packets to the base station periodically along the shortest path which significantly decays the message delay. We call this shortest routing path *the real path*. As soon as a node on the real path receives a packet from the source, it generates a random number $q$ that is uniformly distributed between 0 and 1. If $q < p$, then this node becomes an initial fake source. We use the system parameter $p$ to govern the number of initial fake sources, which has a positive correlation with the length of the real path and is neither smaller than a constant number $N_\alpha$, nor greater than $N_\beta$ (illustrated in detail in Section V). This strategy is applied to achieve a tradeoff between safety period and energy consumption. In Fig. 2, $f_{11}$, $f_{21}$, and $f_{31}$ constitute the first batch of fake sources, i.e., *initial fake sources*. They are produced as soon as the real source sends messages to the base station, which can save energy in WSNs.

### B. Path Extension

After the initial fake sources are determined, each of them will choose a new fake source from its neighbors except the two on the real path to send fake messages to it. Take $f_{11}$ as an example, it chooses $f_{12}$ from its neighbors to send fake event packets to it. While an adversary traces back to $f_{11}$, it may be guided by the fake messages from $f_{12}$ and get farther away from the real path. The routing path of fake messages formed by $f_{11}$ and $f_{12}$ is called a *fake path*.

A fake source continues sending messages along the fake path for a period of $\delta$ and then it selects a new fake source from its neighbors, resulting in a longer fake path. Again, the new fake source cannot be on the real path. Also, the new fake source cannot be neighbor of any node on the real path; otherwise, an adversary residing at this new fake source may be pulled back to the real path by messages routing along it. In other words, the adversary escapes from the fake path. As illustrated in Fig. 2, $f_{12}$ is the neighbor of $f_{11}$, but $f_{13}$ and its successors on that fake path are not neighbors of any sensor node on the real path. Even though $f_{12}$ is a neighbor of some nodes on the real path, the first several sensors on a fake path send messages much faster than the real source and this gives adversaries an extremely small probability to return to the real path if it has been on $f_{12}$ (more details are specified later in this paper). A fake path stops extending if the current fake source cannot select a qualified successor from its neighbors.

The system parameter $\delta$ has important implications for safety period of the real source. As already mentioned in Section III, we assume that an adversary would wait for messages at a spot, and if it overhears no message for some time $\delta'$, it will return to some historical sensor nodes. So, $\delta$ should not be greater than $\delta'$. If an adversary intercepts no message for $\delta'$, it can also return to the base station and then continue to eavesdrop on the whole network. But a sophisticated adversary would record all the historical locations where it has ever stayed. Thus, it can roll back hop-by-hop and return to the real path eventually. In PEM, we assume that an adversary has unlimited storage space and employs the more exquisite attack model.

The locations of fake sources should be considered carefully. Roughly speaking, the fake sources should keep a distance from the real source. Otherwise, an adversary will be pulled toward the real source even though it is on a fake path. If an adversary falls into the visible area [23] of the real source, the monitored object is considered captured. Therefore, the fake paths should not pass through the visible area of the real source. We use system parameter $h_v$, hops from the real source, to represent the visible area. It is guaranteed that a fake path would never get into the visible area of the real source and the initial fake sources are not in the visible area either. This can be achieved by launching a limited flooding by the source as described in Algorithm 2. Nodes that receive this flooding message must not be fake sources.

When a node is chosen to be a fake source, its fake messaging rate can have a significant impact. If the fake source sends messages at a higher rate than the real source, the adversary will be drawn toward the fake source and vice versa. So, as suggested in [12], fake messages should be injected into the network at the same rate as the real messages to gain a balance between safety period and energy consumption. Nonetheless, fake sources are produced dynamically in PEM and the fake paths are extended longer and longer at the same time. So, fake messages along a fake path are designed to be sent fast at first and then the speed is slowed down. Faster delivery rate at the beginning is used to induce adversaries to fall into fake paths, and slower sending rate is to save energy.

**Algorithm 2.** Limited flooding by the source

1: Set the variable *canBeFakeSource* of all the sensor nodes to be *ture*;
2: The source produces a message, $msg\_h_v = h$, and broadcasts it;
3: A sensor node receives a flooding message;
4: **if** $(msg\_h_v > 0)$ **then**
5: $canBeFakeSource = false$;
6: $msg\_h_v = msg\_h_v - 1$;
7: broadcast the modified message;
8: **else**
9: drop the received message;
10: **end if**

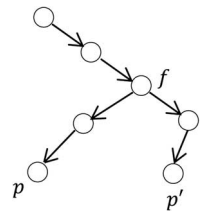



Fig. 3. Path combination.

### C. Combination of Fake Paths

In order to make the length of fake paths as long as possible and save energy, several fake paths may overlap. As shown in Fig. 3, the fake paths $p$ and $p'$ meet at a fake source $f$, and then they converge to a single path. When $f$ receives a message, it will deliver it to the two fake paths. If $f$ is on path $p$ before it joins path $p'$, then the fake messaging rate of $f$ is determined by its location on fake path $p$. Under fake path convergence strategy, the fake sources can be chosen from almost all the sensors in the network. But it is not allowed that a fake source selects its successor on the same fake path(s) with itself. So, in Fig. 3, the fake source $f$ cannot choose its successor on path $p$ or $p'$. Otherwise, it will generate a cycle, which makes an adversary realize that it is entrapped by fake paths through inspecting its recorded locations. A fake path is represented by the initial fake source from which it originates and every fake source memories which path(s) it is on. As depicted in Fig. 3, when the fake path $p'$ encounters $p$ at the fake source $f$, successors of $f$ and itself would modify their memory recording which paths they are on. Now, we can summarize the process of fake source discovery as in Algorithm 3 and the process of fake path extension in Algorithm 4.

**Algorithm 3.** Fake source discovery process

```
1: bool fake_source_discovery_process():
2: // n is the number of neighbors of the current fake source
3: for i = 0 to n do
4:    if (neighbor[i] is on the real path) then
5:       continue;
6:    end if
7:    if (neighbor[i].canBeFakeSource == false) then
8:       continue;
9:    end if
10:   if (the current fake source is an initial fake source) then
11:      new_fake_source = neighbor[i];
12:      if (neighbor[i].is_fake_source == true) then
13:         combine the two fake paths;
14:      else
15:         neighbor[i].is_fake_source = true;
16:         inform neighbor[i] to be a new fake source;
17:         neighbor[i] sends fake messages along the
            extensive fake path;
18:      end if
19:      return true;
20:   else
21:      if (neighbor[i] is neighbor of any node on the real
         path) then
22:         continue;
23:      end if
24:      if (neighbor[i].is_fake_source == true) then
25:         if (neighbor[i] is on the same path with the current
            source) then
26:            continue;
27:         end if
28:         new_fake_source = neighbor[i];
29:         combine the two fake paths;
30:      else
31:         new_fake_source = neighbor[i];
32:         neighbor[i].is_fake_source == true;
33:         inform neighbor[i] to be a new fake source;
34:         neighbor[i] sends fake messages along the
            extensive fake path;
35:      end if
36:      return true;
37:   end if
38: end for
39: return false;
```

**Algorithm 4.** Extension of fake paths

```
1: The current_fake_source sends fake event packets along the
   fake_path for a period of δ;
2: if (fake_source_discovery_process() == false) then
3:    do nothing;
4: else
5:    current_fake_source = new_fake_souce;
6:    fake_path = current_fake_source + fake_path;
7: end if
```

## V. Safety Period Analysis

In this section, we analyze the privacy protection level of PEM. Conventionally, safety period refers to the number of messages sent by the source from the beginning of its generation until it is found by an adversary [9], [12], in which case, we assume that the monitored object stops where it is and the source sends event-reporting messages all the time. In the real scenario, the monitored object will stay for some time $\varphi$ and then move to somewhere else. If $\lambda$ messages should be sent from the source to the base station per unit time during the presence of the monitored object, then $\lambda\varphi$ messages need to be sent. We can assure that the monitored object has left during an adversary's backtracking if $\lambda\varphi < \Delta$ because an adversary need to intercept at least $\Delta$ messages to find the source, where $\Delta$ refers to the safety period of the source. In other words, a monitored object can stay at somewhere safely (not be captured by an adversary) for the time $\Delta/\lambda$ at most. Since $\lambda$ is usually designed to be a constant, larger safety period $\Delta$ of a source means safer monitored object (stay at one place as long as possible and not be captured). It is our goal to guarantee that the monitored object has left when an adversary finally reaches the source. Technically, if a source continues sending messages to the base station, an adversary will locate it eventually. However, a routing policy with security assurance can prolong the safety period of the source and make it *presence secure*.

*Definition 1:* If a monitored object would stay at somewhere for a time period of $\varphi$ without being discovered by any adversary, then it is Presence Secure.

The number of fake sources has a crucial effect on the safety period for all the methods using fake sources. In PEM, the nodes on the path from a source to the base station which are chosen to be fake sources are called *initial fake sources*. They are distraction locations, where adversaries cannot distinguish the correct direction to the source from false one(s). Let $s$ denote the number of initial fake sources on the real path, and then the probability an adversary always selects the correct direction to the source is

$$P_{\text{correct}} = \left(\frac{1}{k+1}\right)^s.$$

In the formula above, $s$ is smaller than $l$, the length of the real path, and $k$ represents the number of fake paths that branch from an initial fake source. In Fig. 2, the value of $k$ is 1 and the corresponding $P_{\text{correct}}$ is $\left(\frac{1}{2}\right)^s$. A greater value of $k$ means much stronger privacy protection and more energy consumption. It is worth noting that the fake sources near the initial fake source on a fake path send messages more often than the real source and it means that an adversary is more likely to trace along a fake path than along the real path. This actually makes the probability of an adversary always selecting the correct direction to the source smaller than $P_{\text{correct}}$ as above.

Let $l$ denote the length of the real path from the source to the base station, and $p$ denote the probability a sensor node on the real path is chosen to be an initial fake source. Then the expectation of $s$ is $E(s) = l \times p$. This suggests that there are fewer fake paths when a monitored object occurs near the base station, which makes the safety period decrease soon. In order

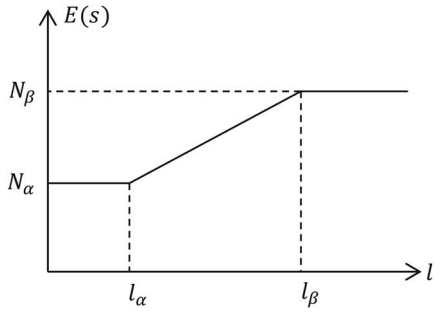



Fig. 4. Relationship of $E(s)$ and $l$.

to balance the energy consumption and safety period, we introduce two thresholds $l_\alpha$ and $l_\beta$, and determine the system parameter $p$

$$p = \begin{cases} \frac{N_\alpha}{l} & l \le l_\alpha, \\ p_0 & l_\alpha < l < l_\beta, \\ \frac{N_\beta}{l} & l_\beta \le l, \end{cases} \quad p_0 = \frac{\frac{N_\beta - N_\alpha}{l_\beta - l_\alpha}(l - l_\alpha) + N_\alpha}{l}.$$

In the formula above, $N_\alpha$ and $N_\beta$ are system parameters. Fig. 4 depicts the relationship between $E(s)$ and $l$. We assume that the distance from source to sink in hops is no smaller than $N_\alpha$. It is obvious that the safety period prolongs with the increasing length of the real path. This is because not only the adversary needs more time tracing the source along the real path, but also the number of delusive fake sources gets larger.

As already mentioned in Section IV, a fake source will launch a fake source discovery process after it continues sending fake event messages for a period of $\delta$. If an adversary is entrapped by a fake path, it will spend $l_f \cdot \delta$ units of time to reach the end of the path, where $l_f$ denotes the length of the fake path. Since the real source sends event packets to the base station periodically until the monitored object leaves it, the report period has a significant influence to the safety period of the source. Similarly, the adversary needs $l \cdot T$ units of time to find the monitored object if no protection measures are used, in which $l$ denotes the length of the real path, and $T$ denotes the event-reporting period. So, during the time when an adversary is entrapped by a fake path, $l_f \cdot \delta/T$ messages are sent by the real source.

From the analysis above, we can see that there are two important factors affecting the safety period of a real source, which are $l_f$, length of the fake path, and the ratio value, $\delta/T$. Note that $l$ has an important impact on the safety period, but where the monitored object occurs is absolutely a random event. For a fixed event-reporting period $T$, greater values of $l_f$ and $\delta$ make longer safety period of the real source. There are two strategies an adversary can use when it gets to the end of a fake path. One simple way is going back to the base station and overhearing the whole network from the origin. But a sophisticated adversary may store the information of all historical locations where it has ever eavesdropped. This helps it to roll back hop-by-hop and return to the real path eventually. We assume that an adversary has unlimited storage for historical information. So, let $\delta'$ be the time during which an adversary overhears at one location, it actually misses $(l_f \cdot \delta/T + l_f \cdot \delta'/T)$ messages from the time it enters a fake path to the time it returns to the real path. Since $\delta < \delta'$, $2l_f \cdot \delta'/T$ safety period is achieved at most if an adversary is entrapped by a fake path. So, if an adversary is entrapped by just one fake path, then the safety period is $S_{\text{theoretical}} = (2l_f \cdot \delta'/T + l)$. It can be seen that different ratios of $\delta'/T$ will lead to different protection strengths of PEM. A greater value of $\delta'$ means longer time of wait and entrapping on a fake path. On the contrary, a smaller value of $\delta'$ means more unnecessary back rollings. So, the adversary may choose the value of $\delta'$ carefully.

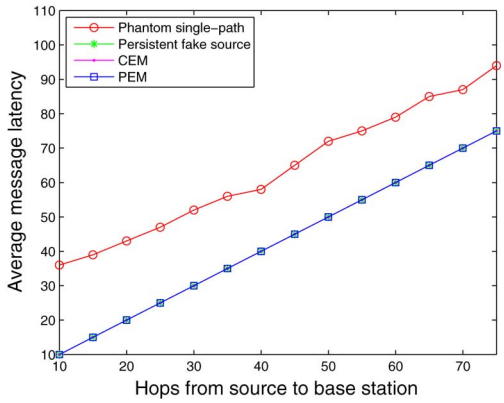



Fig. 5. Comparison of average message latency.

## VI. Experiment and Performance Evaluation

In this section, we will present the performance of PEM in terms of safety period and energy consumption. We do the simulation experiments in OMNeT++, an object-oriented modular discrete event network simulation framework. We compare the performance of PEM with phantom single-path routing technique proposed in [12] and the other two representative methods of fake messaging, i.e., *persistent fake source routing strategy* [12] and *CEM* [19]. In the methods of persistent fake source, CEM and PEM, messages from the source are routed to the base station along the shortest path, which gives them minimum latency (proportional to the distance between the source and base station) and high delivery reliability (almost 100%). In phantom routing, messages are first sent to some other nodes by the source through random walks and then delivered to the base station along the shortest path, so its message latency is slightly higher than the other three methods. The comparison in regard to message latency is shown in Fig. 5, in which the length of random walks is set to be $H_w = 15$ as in [12] and the latency is denoted by the average hop counts that messages pass through from the source to the base station. For each distance, 10 random sources are selected and the average results are presented.

### A. Configuration of Simulation Experiment

To make convenient comparisons between PEM and other methods, we inherit some configurations of simulating environment in [12] and assume that the radio range of sensor nodes is 100 m. We do the simulation experiments in a

5000 m $\times$ 5000 m area with 5000 sensor nodes uniformly distributed. Given a WSN with a fixed size, the network performance is better if the base station is located in the centroid of the region due to smaller message delivery latency. But it also makes adversaries track the source more easily. There is usually a tradeoff between safety and network performance to decide where to locate the base station in practical applications. However, it makes little difference as to performance comparison between PEM and other routing strategies no matter where the base station is. So, we locate the base station in the middle of one edge of the square to make the distance between the base station and a randomly occurred source as long as possible. Theoretically speaking, the longest distance is approximately 55 hop counts (77 hops in the actual simulation experiment, which is sufficient and almost the same as in [12]). We can achieve the same purpose with more sensors distributed in a larger area (e.g., 10 000 m $\times$ 10 000 m area with 20 000 sensors and a centroid-located base station), but that is too large for a WSN and also a huge burden on our simulation program. For a sensor node, the number of its neighbors is approximately 6.28 ($\pi \times 100^2 \div 5000^2 \times 5000 \approx 6.28$). In practice, all the sensor nodes can determine their shortest paths to the base station and identify their neighbors through a message flooding from the base station after network deployment.

When a sensor is randomly selected to be a source, it immediately launches a limited flooding as shown in Algorithm 2, and sends messages along the shortest path to the base station every $T$ units of time. The actual value of $T$ depends on the rate of message generation and has no effect on the simulation results. It is assumed to be 30 s in the simulation experiment. After that, fake paths emanated from the initial fake sources on the real path are generated. As mentioned in Section IV-B, fake messages along a fake path are designed to be sent fast at first and then the speed is slowed down. Faster delivery rate at the beginning is used to induce adversaries to fall into fake paths, and slower sending rate is to save energy. So, we let the first five fake sources on the fake path send fake messages twice faster than the real source and other fake sources on it inject fake messages four times slower than the source. The parameters are carefully and empirically selected in the simulation to induce adversaries into fake paths and make it hard to be aware of. Since a source is 55 hops away from the base station at most, we set $l_\alpha$, $l_\beta$, $N_\alpha$, $N_\beta$ to be 15, 45, 8, and 20, respectively, to balance the safety period and energy overhead (see Fig. 4). The meanings of these four parameters have already been described in Section V.

### B. Safety Period

We compare safety period of the four methods in Fig. 6. For each distance from the source to the base station, several qualified sources may exist, so we calculate and present the average safety period. The visible area of a source $h_v$ is set to be 1 hop, namely 100 m, to enlarge the expanding range of fake paths. As mentioned before, the length of random walks in phantom routing technique is set to be $H_w = 15$. Although a greater value of $H_w$ makes better performance in terms of safety period, higher message latency and more energy consumption will be incurred. Again, we consider the tradeoff between privacy protection and energy consumption when choosing the system parameters. As for PEM, an adversary would stay and eavesdrop at one spot for $\delta' = 2$ min before its back rolling, which is four times the period of a source reporting messages to the base station. The influence of $\delta'/T(>1)$ on safety period is depicted in Fig. 7, in which $T$ represents the event-reporting period and the source is 30 hops away from the base station. As we can see, the safety period of a source is approximately proportional to $\delta'/T$, which is consistent with the analysis in Section V. The result is intuitional because the longer an adversary stays at fake sources, the safer a source is.

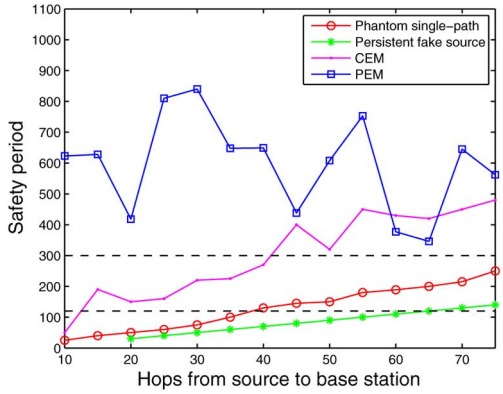



Fig. 6. Comparison of safety period.

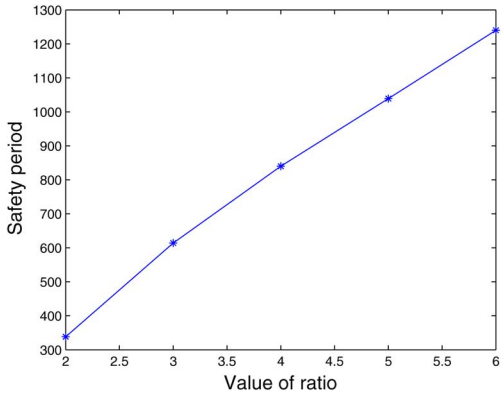



Fig. 7. Influence of $\delta'/T$ on safety period.

As shown in Fig. 6, the performance of PEM is significantly better than the other three methods. On average, the safety period of PEM is 6.5 times larger than that of phantom single path and the ratio of both is as high as 25 when the source is 10 hops away from the base station. As Fig. 6 indicates, PEM performs quite well even if the source is close to the base station while the safety period of the other three methods is not satisfying in this case. Even though the average safety period of PEM is 1.98 times larger than that of CEM, the performance of PEM is far better than CEM when the distance between the source and base station is less than 45 hop counts. The persistent fake source routing strategy has the worst performance because a position-unchanged fake source will be easily pinpointed by

the adversary. An eavesdropper would return to the path routing messages from the source to the base station after he discovers and identifies the fake source. On the contrary, the dynamically generated fake sources in PEM can induce an adversary farther away form the source without being aware of. From Fig. 6 we can see that, if a monitored object would stay at someplace for 1 h, i.e., $\Delta = 1$ h (corresponding to the safety period of 120), then it is *presence secure* only if its distance from the base station is no less than 40 hops when using phantom single path as routing strategy, and when $\Delta$ increases to 2.5 h (corresponding to the safety period of 300), the object is sure to be found wherever it occurs. On the contrary, the monitored object is *presence secure* in both cases if PEM is deployed in the WSN. According to the analysis in Section V, the safety period of PEM is closely related to the length of fake paths, but which fake paths the adversary may get into is uncertain and this makes the performance of PEM fluctuate much.

### C. Energy Consumption

There is always a tradeoff between security and energy consumption, no matter what kind of routing policy is employed in the sensor network. It is optimal when the messages are routed from the source to the base station along the shortest path if we only take latency and energy consumption into account. However, an adversary will easily trace back to the source and find the monitored asset in this situation. In phantom single-path routing, random walk is introduced to secure the source which lengthens the routes and consumes more energy. In PEM, fake paths are generated and fake messages are injected into the network to entrap adversaries. Both methods trade energy for security of the monitored objects. Since energy is very limited in sensor networks and affects network lifetime, routing strategy should be designed carefully to consume as less energy as possible. In this section, we will analyze the energy consumption of PEM.

There are two flooding processes when running PEM. The first one is launched by the base station soon after network deployment and it aims to determine the shortest path to the base station and identify neighbors of every node. The other is launched by the source to prescribe a limit of its visible area. The energy consumption of flooding is definitely very high, especially in a large-scale sensor network. But the first flooding process initiated by the base station is necessary for any routing protocol, and the other one is a limited flooding which only involves $6.28h_v^2$ sensor nodes. In PEM, $h_v$ is set to be 1, so an extremely small part of node participate in the limited flooding. Hence, it is an acceptable energy consumption of the two flooding processes in PEM.

Consider a fake path $p$ with length $l$. Let $T$ be the event-reporting period of the real source. According to the experimental parameter settings, the first five fake sources on $p$ send fake messages every $T/2$, and the subsequent fake sources send messages every $4T$. Take Fig. 2 as an example, the fake source $f_{12}$ sends fake messages to the initial fake source $f_{11}$ for a period time of $\delta$, then $2\delta/T$ fake messages are sent by $f_{12}$. The fake source $f_{13}$ sends fake messages along the fake path formed by $f_{13}$, $f_{12}$, and $f_{11}$, so $2 \cdot 2\delta/T$ fake messages are sent during the time when $f_{13}$ is the current fake source. Therefore, given a fake path of length $l$

$$\frac{2\delta}{T}(1+2+3+4+5) + \frac{\delta}{4T}\sum_{i=6}^{l} i = \frac{\delta}{T}\cdot\left(30 + \frac{(6+l)(l-5)}{8}\right)$$

fake messages are sent during its lifetime. Let $l'$ be the average length of all the fake paths, then the total energy consumed by all the fake paths is given by

$$s \cdot \frac{\delta}{T}\cdot\left(30 + \frac{(6+l')(l'-5)}{8}\right)\cdot(E_{\text{send}} + E_{\text{receive}})$$

where $s$ denotes the quantity of fake paths, and $(E_{\text{send}} + E_{\text{receive}})$ denotes the energy consumption delivering a message from one node to its neighbors. If $\delta <= 4T$, every fake source except the first five on path $p$ sends only one fake message when being added to path $p$. In this case, the total energy consumed by all fake paths is

$$s \cdot \left(\frac{30\delta}{T} + \frac{(6+l')(l'-5)}{2}\right)\cdot(E_{\text{send}} + E_{\text{receive}}).$$

Meanwhile, the total energy consumed by messages sent by the source is (SafetyPeriod). $l_{\text{RealPath}} \cdot (E_{\text{send}} + E_{\text{receive}})$.

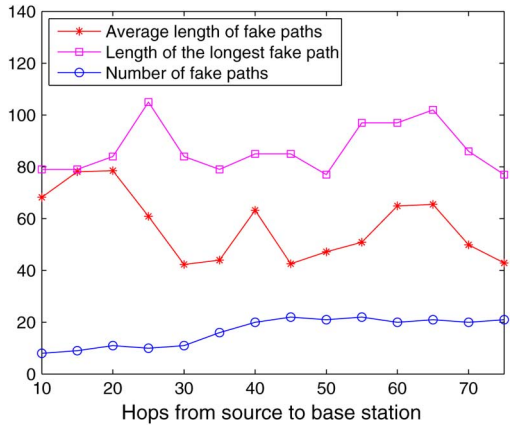



Fig. 8. Variation of $s$ and $l'$.

Fig. 8 shows how the change of distance between the source and base station affects $s$ and $l'$. The variation of $s$ is almost the same with the curve shown in Fig. 4. The average length of fake paths decreases with the increasing distance between the source and base station. This is because longer real paths kind of partition the network into two parts and limit the expanding range of fake paths. The average length of fake paths decreases from 78 to 42 hops when the distance between the source and base station increases from 15 to 75 hops.

The number of fake messages injected by all fake paths, i.e., $C_f = s.(30\delta/T + (6+l')(l'-5)/2)$ and messages from the source, i.e., $C_r = l_{\text{RealPath}} \cdot$ (SafetyPeriod) are depicted in Fig. 9. From the above formula, we can see that $C_r$ almost completely relies on the safety period of the source, since (SafetyPeriod) is usually considerably larger than $l_{\text{RealPath}}$. So the curve of $C_r$ should have almost the same form with that of

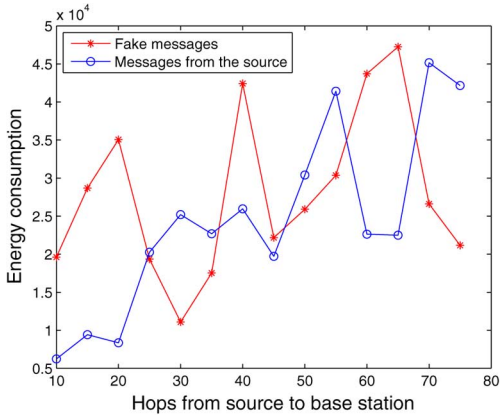



Fig. 9. Energy consumption.

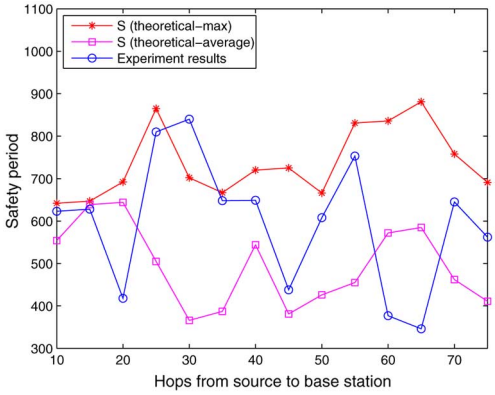



Fig. 10. Comparison of safety period.

(SafetyPeriod). The curve "messages from the source" in Fig. 9 depicts $C_r$ and the curve "experiment results" in Fig. 10 depicts (SafetyPeriod) (also illustrated in Fig. 6). We can see that the two curves are indeed similar, following the same trends. Similarly, from the above computational formula of $C_f$, we can see that it is approximately proportional to $l'^2$ for a given $\delta/T$ and little changeable $s$. The curve "fake messages" in Fig. 9 depicts $C_f$ and the curve "average length of fake paths" in Fig. 8 depicts $l'$. We can see that the two curves follow the same trends except the amplitude of variation, which is consistent with the above analysis. On average, the energy consumed by fake messages is 1.57 times that consumed by event messages. The maximum ratio of both is 4.2 when the source is 20 hops away from the base station, and the minimum is 0.44 while the source occurs 30 hops away from the base station. It is worth noting that the actual energy consumed by fake messages is less than the value calculated above due to paths combination. So, the energy overhead of PEM is completely acceptable.

As we have mentioned in Section V, if an adversary is entrapped by a fake source, it will miss $2l_f \cdot \delta'/T$ messages sent by the source. Since the first five fake sources on a fake path send fake messages faster than the source, an adversary is certain to be misled by at least one fake path, and the safety period is $S_{\text{theoretical}} = (2l_f \cdot \delta'/T + l)$ in this case. Taking $l_f$ as the length of the longest fake path and $l$ as the average length of all the fake paths, we can compare $S_{\text{theoretical}}$ with the experiment results shown in Fig. 10. We can see that it is hard to exactly

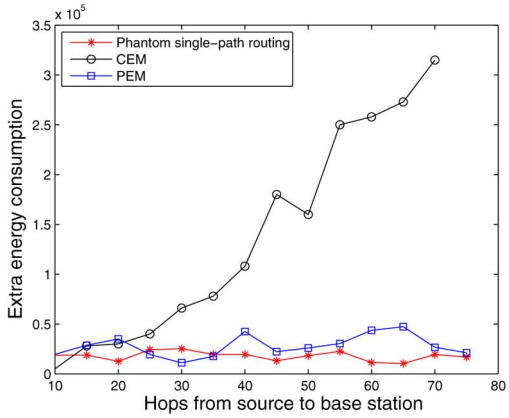



Fig. 11. Comparison of energy consumption.

entrap an adversary in the longest fake path and it is also hard for the adversary to escape from fake paths since it may jump from one fake path to another while being misled.

According to the above analyses, for PEM, it introduces $C_f = s \cdot (30\delta/T + (6 + l')(l' - 5)/2)$ extra transmissions to the shortest path between the source and the base station. For phantom single-path routing, it incurs at most $C_p = 2h_{\text{walk}} \cdot (\text{SafetyPeriod})$ extra transmissions, where $h_{\text{walk}}$ denotes the length of random walks [12]. Based on the analysis in [19], the extra transmissions incurred by CEM is $C_e = l_{\text{shortest}} \cdot l_{\text{loop}} \cdot (\text{SafetyPeriod})$, where $l_{\text{shortest}}$ denotes the length of the shortest path between the source and the base station, $l_{\text{loop}}$ denotes the average length of loops deployed in the WSN. Fig. 11 shows the comparison of the three in regard to extra message transmissions. We can see that the extra energy consumption of PEM is more than phantom single-path routing for most of the time. On average, the extra energy consumption of PEM is 1.80 times larger than that of phantom single-path routing technique. The largest and smallest ratios of the two are 4.55 and 0.44, respectively. However, from Fig. 6 (Section VI-B), we can see that the performance of PEM is significantly better than phantom single-path routing. On average, the safety period of PEM is 6.5 times larger than that of phantom single path, and the ratio of both is as high as 25 when the source is 10 hops away from the base station. According to the performance analysis in [19], there is very small performance difference of CEM when the average length of loops is not less than 10. So we choose $l_{\text{loop}}$ to be 10 in the simulation experiment of CEM. Still, it consumes much more energy than PEM and phantom single-path routing. Intuitively, even if an adversary is entrapped by loops deployed in CEM, he will eventually return to the actual routing path of event-reporting messages after tracing around a circle. Therefore lots of loops have to be deployed in the WSN to achieve a relatively high level of security. Hence, deploying fake paths in the WSN rather than loops is a smarter choice and incurs less energy consumption. As for *persistent fake source* routing strategy, it has the worst performance with respect to safety period and consumes the least energy. The extra energy consumption of persistent fake source is about the same as the energy used to deliver messages from the real source. The energy consumption of messages from the source has already

depicted in Fig. 9. In summary, it is worthwhile trading that excessive energy in PEM for stronger source privacy protection. We provide a great alternative for making deliberate choice to balance between source privacy protection and energy consumption in practical applications.

## VII. Conclusion

Source-location privacy protection is a significant security property of sensor networks used to collect information about monitored objects in military or endangered species-monitoring applications. Secure routing protocols should be designed to prevent adversaries from finding out the source through hop-by-hop backtracking. To this aim, PEM is proposed to provide strong protection for source-location privacy where fake sources are generated dynamically and several fake paths are formed and extended in the network. Adversaries would be induced farther away from the source if they are entrapped by some of the fake paths. It performs quite well even though an object occurs near the base station. The theoretical and simulation results show that PEM can provide strong source-location privacy protection with minimal message latency and acceptable overhead. As future work, we will investigate different defense-attack models under multiple and mobile sources.

## References


[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[2] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: A link layer security architecture for wireless sensor networks," in *Proc. 2nd Int. Conf. Embedded Netw. Sensor Syst. (SenSys'04)*, New York, NY, USA, Nov. 2004, pp. 162–175.

[3] J. Deng, R. Han, and S. Mishra, "Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks," in *Proc. Int. Conf. Depend. Syst. Netw. (DSN'04)*, Florence, Italy, Jun. 2004, pp. 637–646.

[4] J. Deng, R. Han, and S. Mishra, "Countermeasures against traffic analysis attacks in wireless sensor networks," in *Proc. 1st IEEE Conf. Security Privacy Emerg. Areas Commun. Netw. (SecureComm'05)*, Athens, Greece, 2005, pp. 113–126.

[5] (2005). The Free Haven project [Online]. Available: http://freehaven.net/anonbib/date.html

[6] D. Chaum, "Untraceable electronic mail, return address, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–88, Feb. 1981.

[7] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in *Proc. 1st IEEE Int. Workshop Sensor Netw. Protocols Appl.*, Anchorage, AK, USA, May 2003, pp. 113–127.

[8] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran, "Toward a statistical framework for source anonymity in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 2, pp. 248–260, Feb. 2013.

[9] C. Ozturk, Y. Zhang, and W. Trappe, "Source-location privacy in energy-constrained sensor network routing," in *Proc. 2nd ACM Workshop Security Adhoc Sensor Netw. (SASN'04)*, Washington, DC, USA, Oct. 2004, pp. 88–93.

[10] Y. Li and J. Ren, "Source-location privacy through dynamic routing in wireless sensor networks," in *Proc. 29th IEEE Int. Conf. Comput. Commun. (INFOCOM'10)*, San Diego, CA, USA, Mar. 2010, pp. 1–9.

[11] A. Gurjar and A. R. B. Patil, "Cluster based anonymization for source location privacy in wireless sensor network," in *Proc. Int. Conf. Commun. Syst. Netw. Technol. (CSNT'13)*, Gwalior, India, Apr. 2013, pp. 248–251.

[12] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *Proc. 25th Int. Conf. Distrib. Comput. Syst. (ICDCS'05)*, Columbus, OH, USA, Jun. 2005, pp. 599–608.

[13] Y. Xi, L. Schwiebert, and W. Shi, "Preserving source location privacy in monitoring-based wireless sensor networks," in *Proc. 20th Int. Parallel Distrib. Process. Symp. (IPDPS'06)*, Rhodes Island, Greece, Apr. 2006, pp. 162–169.

[14] K. Mehta, D. Liu, and M. Wright, "Location privacy in sensor networks against a global eavesdropper," in *Proc. IEEE Int. Conf. Netw. Protocols (ICNP'07)*, Beijing, China, Oct. 2007, pp. 314–323.

[15] M. Shao, Y. Yang, S. Zhu, and G. Cao, "Towards statistically strong source anonymity for sensor networks," in *Proc. 27th IEEE Int. Conf. Comput. Commun. (INFOCOM'08)*, Phoenix, AZ, USA, Apr. 2008, pp. 51–59.

[16] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao, "Towards event source unobservability with minimum network traffic in sensor networks," in *Proc. 1st ACM Conf. Wireless Netw. Security (WiSec'08)*, New York, NY, USA, Mar. 2008, pp. 77–88.

[17] H. Wang, B. Sheng, and Q. Li, "Privacy-aware routing in sensor networks," *Comput. Netw.*, vol. 53, no. 9, pp. 1512–1529, Jun. 2009.

[18] K. Pongaliur and L. Xiao, "Maintaining source privacy under eavesdropping and node compromise attacks," in *Proc. 30th IEEE Int. Conf. Comput. Commun. (INFOCOM'11)*, Shanghai, China, Apr. 2011, pp. 1656–1664.

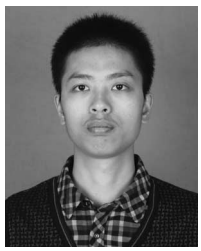
[19] Y. Ouyang, Z. Le, G. Chen, J. Ford, and F Makedon, "Entrapping adversaries for source protection in sensor networks," in *IEEE Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM'06)*, Buffalo–Niagara Falls, NY, USA, Jun. 2006, pp. 25–34.

[20] K. Mehta, D. Liu, and M. Wright, "Protecting location privacy in sensor networks against a global eavesdropper," *IEEE Trans. Mobile Comput.*, vol. 11, no. 2, pp. 320–336, Feb. 2011.

[21] H. Park, S. Song, B. Y. Choi, and C. T. Huang, "PASSAGES: Preserving anonymity of sources and sinks against global eavesdroppers," in *Proc. 32th IEEE Int. Conf. Comput. Commun. (INFOCOM'13)*, Turin, Italy, Apr. 2013, pp. 210–214.

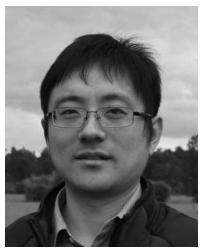
[22] U. Srimongkolpitak and Y. Yang, "Sensor source location privacy based on random perturbations," in *Proc. 8th Int. Conf. Collab. Comput. Netw., Appl. Worksharing (CollaborateCom'12)*, Pittsburgh, PA, USA, Oct. 2012, pp. 500–507.

[23] W. Wang, L. Chen, and J. Wang, "A source-location privacy protocol in WSN based on locational angle," in *Proc. IEEE Int. Conf. Commun. (ICC'08)*, Beijing, China, May 2008, pp. 1630–1634.

**Wei Tan** received the B.Sc. degree in computer science and technology from Sun Yat-sen University, Guangzhou, China, in 2012, and is currently working toward the Master's degree in computer science and technology from Tsinghua University, Beijing, China.

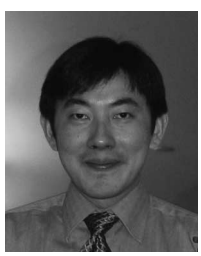
His research interests include wireless networks and wireless sensor networks.



**Ke Xu** (M'02–SM'09) received the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2001.

He is currently a Full Professor with Tsinghua University. He is also currently a Visiting Professor with the University of Essex. He has authored or coauthored more than 100 technical papers. He holds 20 patents in the study of next-generation Internet, P2P systems, Internet of Things (IoT), network virtualization, and optimization.

Dr. Xu is a member of the ACM. He has been a Guest Editor for several special issues of IEEE and Springer journals.



**Dan Wang** (S'05–M'07–SM'13) received the B.Sc. degree from Peking University, Beijing, China, in 2000, the M.Sc. degree from Case Western Reserve University, Cleveland, OH, USA, in 2004, and the Ph.D. degree from Simon Fraser University, Burnaby, BC, Canada, in 2007, all in computer science.

He is currently an Associate Professor with the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong. His research interests include wireless sensor networks, Internet routing, and applications.