On protecting end-to-end location privacy against local eavesdropper in Wireless Sensor Networks

Honglong Chen\textsuperscript{a,b,*}, Wei Lou\textsuperscript{b,c}

\textsuperscript{a} College of Information and Control Engineering, China University of Petroleum, Qingdao, PR China
\textsuperscript{b} Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong
\textsuperscript{c} The Hong Kong Polytechnic University Shenzhen Research Institute, Shenzhen, PR China

A R T I C L E   I N F O

Article history:
Received 11 March 2013
Received in revised form 13 January 2014
Accepted 21 January 2014
Available online xxxx

Keywords:
Local eavesdropper
Location privacy
Wireless sensor networks

A B S T R A C T

Wireless Sensor Networks (WSNs) are often deployed in hostile environments to detect and collect interested events such as the appearance of a rare animal, which is called event collection system. However, due to the open characteristic of wireless communications, an adversary can detect the location of a source or sink and eventually capture them by eavesdropping on the sensor nodes’ transmissions and tracing the packets’ trajectories in the networks. Thus the location privacy of both the source and sink becomes a critical issue in WSNs. Previous research only focuses on the location privacy of the source or sink independently. In this paper, we address the importance of location privacy of both the source and sink and propose four schemes called forward random walk (FRW), bidirectional tree (BT), dynamic bidirectional tree (DBT) and zigzag bidirectional tree (ZBT) respectively to deliver messages from source to sink, which can protect the end-to-end location privacy against local eavesdropper. Simulation results illustrate the effectiveness of the proposed location privacy protection schemes.

1. Introduction

Recent advancement in wireless communications and Micro-Electro-Mechanical Systems (MEMS) has enabled the development of low-cost Wireless Sensor Networks (WSNs), which are made up of a number of sensor nodes that are self-organized for various applications, such as mobile target detection [1], earthquake monitoring [2], and habitat monitoring [3]. In these applications, sensor nodes are deployed to detect the existence of an interested event, such as the appearance of a rare animal. The sensor nodes that detect the occurrence of the interested event will send the detection information to a sink (or base station) by multi-hop wireless communications. Such kind of systems is called event collection system [4], which is one of the important applications in WSNs.

Due to the open characteristic of wireless communications, it is not difficult to attack wireless sensor networks with the goal of either obtaining confidential data or simply disrupting the normal operations of the WSN applications [5–7]. In either case, they may involve threats to one of the following two types of WSN privacy, content privacy and contextual privacy [8]. The former refers to the confidentiality of the content of the packets passing between the nodes in the network. This is usually guaranteed by using methods of encryption and authentication [9]. The latter refers to the confidentiality of information about traffic patterns in the network, which may be used by adversaries to disrupt the network. The location privacy, i.e., the confidentiality of the location of either source, sink, or both, is a kind of contextual privacy.

* Corresponding author at: College of Information and Control Engineering, China University of Petroleum, Qingdao, PR China. Tel.: +86 13573861376.
E-mail addresses: honglongchen1984@gmail.com, chenhlpupc.edu.cn (H. Chen), csweilou@comp.polyu.edu.hk (W. Lou).
To illustrate how information about traffic patterns in a network might be exploited by an adversary, we consider a habitat monitoring application called “Panda-Hunter” [8] as shown in Fig. 1, in which a typical WSN is deployed to monitor the appearance of the pandas in the wild field. There is a central controller (sink in Fig. 1) and several pandas in the monitoring field. The sensor nodes which detect the appearance of the pandas will act as source nodes and will send the monitoring packets to the central controller via multi-hop wireless communications. The central controller can then analyze the life habit of the pandas after receiving the monitoring packets or further send the data to a powerful computer for more complex analysis. The scenario is obviously unsafe as the hunter (adversary in Fig. 1) is easily able to either locate a source by back tracing the packet transmissions hop-by-hop to capture the panda or locate the sink by following the flow of packet transmissions to threaten the central controller of the monitoring system. The challenge in this scenario is essentially to protect the end-to-end location privacy rather than merely protect the source or sink location privacy. Thus the end-to-end location privacy protection is a crucial contextual privacy problem in WSNs.

In this paper, we propose four end-to-end location privacy protection schemes to deliver messages from source to sink, which can protect against local eavesdropper that might break the location privacy of a source or sink, i.e., the end-to-end location privacy. Hereafter in this paper, we use terms eavesdropper and adversary interchangeably. The proposed four location privacy protection schemes are called forward random walk (FRW), bidirectional tree (BT), dynamic bidirectional tree (DBT) and zigzag bidirectional tree (ZBT) respectively. In the forward random walk scheme, every node relays a received packet to a node randomly chosen from its forward neighbors whose hop count to the sink is not larger than its own. To enhance the location anonymity of the source and sink, a tree topology is employed at the two ends of the delivery path respectively in the bidirectional tree scheme. In the dynamic bidirectional tree scheme, branches of the trees are generated dynamically to further improve the performance. However, in the bidirectional tree scheme, real messages are delivered along the shortest path, making it possible for the eavesdropper to infer the location of the source or sink by extending the line of the shortest path. To solve this potential threat, a proxy source and a proxy sink are adopted in the zigzag bidirectional tree scheme, which prevents the adversary from inferring the location of the source or sink easily.

The main contributions of this paper can be summarized as follows:

- We address the importance of simultaneously protecting the location privacy of both the source and sink;
- We propose four schemes to deliver messages from source to sink, which can protect the end-to-end location privacy against the local eavesdropper;
- We demonstrate the effectiveness of the proposed schemes through TOSSIM-based simulations.

The rest of this paper is organized as follows. Section 2 reviews the existing location privacy preserving techniques. Section 3 proposes the system scenario, adversary model and the metrics of location privacy protection. Section 4 describes our proposed four location privacy protection schemes. Section 5 evaluates the performance of the proposed schemes under the TOSSIM platform. Finally, Section 6 concludes this paper and puts forward the future work.

2. Related work

Location privacy protection [8,10–12] for WSNs has been a hot research topic during the past years. Most of existing schemes have addressed the location privacy protection of the source or sink independently:

- Source location privacy protection: In [8,13], a source location privacy protection scheme was proposed, which uses the “Panda-Hunter” problem as an application scenario for monitoring-oriented sensor networks where the location privacy is important. The Phantom routing protocol makes use of a random walk to prevent attackers identifying the source. Xi et al. [10] proposed a two-way random walk routing protocol (from both the source and sink) called greedy random walk,
which can reduce the opportunity for an eavesdropper to collect the location information. PRLA [14] protects the source location privacy by using so-called inclination angles to ensure that every random walk gets away from the region close to the source, which enhances the source location privacy. In [15], loops are created in the network. The adversary has to go around these loops, thereby being led away from the real path, which guarantees a high privacy. A suboptimal privacy routing scheme called WRS was proposed in [16] to protect the source location privacy by distributing message flows to different disjoint routes. It also formulated the performance bound for any routing scheme. Li and Ren [17,18] protected the source location privacy through a two-phase routing process. In the first phase, the packet travels randomly through the intermediate nodes before it is routed to a ring node. Then the second phase is triggered and the packet is mixed with other packets through a network mixing ring (NMR). In [19], two techniques called periodic collection and source simulation were proposed. In the periodic collection, every node sends messages periodically, making the network be $n$-anonymous, which means that the probability that the eavesdropper can identify a specific node in the network is $1/n$. In the source simulation, some faked sources send packets to mislead the eavesdropper away from the real source, which can well balance the trade-offs between privacy, communication cost and latency. Four schemes named naive, global, greedy and probabilistic were proposed in [20] to provide location privacy against laptop-class attacks. Yang et al. [21] proposed to use proxies to protect the source location of an event. A prototype of this scheme was implemented on Mica2 motes. In [22] FitProbRate was proposed, which first adopts the statistically strong source anonymity to reduce the latency efficiently.

Sink location privacy protection: In [11], Deng et al. proposed a base station privacy scheme against the traffic-rate analysis attack by randomly delaying the transmission time of each packet. They also proposed in [23] to defend against the traffic analysis attacks. They first designed a multi-path routing to multiple destination base stations to provide intrusion tolerance against isolation of a base station. They also proposed anti-traffic analysis strategies to disguise the location privacy of the base station. LPR [12] provides receiver location privacy against the packet tracing attacks. In LPR, the directions of both incoming and outgoing from a sensor node are uniformly distributed, which makes it difficult for the adversary to ascertain the direction of the sink. Fake messages are also injected to get a longer safety period with the cost of increasing the energy consumption in the network. In [24], an enhancing sink location privacy protection scheme against the global eavesdropper was proposed, which can achieve $k$-anonymity in the network so that at least $k$ entities are indistinguishable to the nodes around the sink. SLPP [25] was proposed to protect the sink location privacy by injecting fake messages, in which the number of fake message is dependent on the number of the node’s children so that it will not seriously affect the network’s lifetime. Ngai et al. [26] proposed to provide location privacy for the mobile sinks, in which the sinks moves along some random paths to collect data from local nodes so that the attackers cannot predict their locations and movements.

The common drawback of the above schemes is that none of them considers the location privacy of both the source and sink simultaneously. The location privacy issue of the source and destination has been discussed in [27,28] for the shortest path computation in location-based services (LBSs). In [27], an approach called information leakage-aware cloaking (ILC) was proposed to preserve the location privacy of the querying users. The basic idea is to cloak the anonymous set by using multiple anonymous spatial regions, each of which contains more than $m$ users. ILC can achieve $k$-anonymity for the querying users. In [28], several advanced PIR-based methods were proposed to guarantee no information leakage by using the same query plan for all queries to achieve an identical chronological order of the page retrievals and adding fake pages into the query procedure to make the number of page retrievals in the various files be identical. Consequently, the adversary cannot distinguish any queries. However, these schemes focus on the privacy protection of the LBS system which is usually a database-driven system and is not suitable for the WSNs. This motivates us to design protection schemes that aim to protect the location privacy of both the source and sink for the WSNs, which is particularly important for applications such as the habitat monitoring system in Fig. 1.

3. Problem statement

In this section, we will describe a generic system scenario in which a WSN is potentially threatened by a particular adversary which seeks to break the location privacy of the source or sink. After that we will describe the adversary model in detail. Finally we will define the metrics of safety period, end-to-end latency and energy consumption, which will be used to evaluate the location privacy protection schemes. The major goal of these protection schemes is to achieve a high safety period with a low end-to-end latency and energy consumption.

3.1. System scenario

We consider a WSN-based monitoring system called “Panda-Hunter” as shown in Fig. 1 which is comprised of a sink node and many sensor nodes. The panda is the target of the monitoring system. It may appear at certain time, move around for a while and then move into its cave for another while, and so on. The sensor node which detects the appearance of a panda will act as a source node and periodically send the monitoring packets to the sink. If the source node cannot detect the panda for a certain period, it will stop sending the monitoring packets to the think. The monitoring system may be threatened by the hunter, a particular adversary, who seeks to break the source location privacy by back-tracking the packet transmissions hop-by-hop to capture the panda or break the sink location privacy by following the flow of packet transmissions to find the sink. Thus, it is equivalently important to protect the location privacy of the source and sink simultaneously to guarantee the
security of both the panda and the sink when delivering messages in such systems. Notice that the adversary has to capture
the source or sink before the panda disappears, or he has to wait until the next time the panda appears.
We assume that all the sensor nodes are identically configured, i.e., they have the same capability and communication
range, which is denoted as \( R \). Two sensor nodes can communicate with each other when their distance is less than \( R \). The sink
is assumed to have greater capability than that of the sensor nodes. After deploying the WSN, the sink initiates a flooding,
which provides each sensor node three types of information: (1) its minimum hop count to the sink; (2) its neighboring
nodes; (3) the minimum hop count from each neighboring node to the sink. Whenever a sensor detects the appearance of a
panda, it will send monitoring packets to the sink periodically with an interval \( T_s \) using certain routing strategies.

3.2. Adversary model

We assume the adversaries to be equipped with some powerful devices such as spectrum analyzer, which can be used to
localize the sender of a transmitted packet [12]. Typically, adversaries against the contextual privacy can be classified into
two categories: local (or mote-class) adversaries and global (or laptop-class) adversaries [29]. Local adversaries are assumed to
have a local view of the network traffic, which means that they can only overhear the packets within the transmission range.
Global adversaries acquire a global view of the network traffic, based on which they can eavesdrop on every transmitted
packet in the network and then localize a specific node. Though the global adversaries can seriously threaten the network,
they are very difficult to be implemented, especially in a large-scale WSN. In this paper, we focus on the location privacy
protection of both the source and sink against the local adversary.

We define the characteristics of a local adversary as follows, some of which are borrowed from the “Panda-Hunter”
model [8]:
1. The adversary randomly walks in the network until it overhears a packet transmitted by some node.
2. The adversary cannot decrypt the content of the packet as every packet is encrypted. Also, the adversary will not actively
interfere with the proper function of the network. For example, the adversary will not modify packets in transit, alert
the routing path or destroy sensor devices, since there may exist intrusion detection mechanisms to detect these active
malicious behaviors.
3. The adversary is equipped with powerful devices, such as directional antenna and spectrum analyzer, which can be used
to measure the arriving angle and the received signal strength of a message. Based on these measurements, the adversary
can identify the location of the immediate sender.
4. The adversary randomly decides whether to trace the source (to capture the panda) or sink (to destroy the whole
monitoring system) when it intercepts a transmitted packet. The adversary can localize the source (or sink) of a traffic
flow by analyzing the traffic flow and tracing back (or forth) hop-by-hop. When the distance to the target (the source or
sink) is short enough, the adversary can detect the target directly.
5. The movement of the adversary is far slower than the transmitting speed of a packet in the network. Therefore, the
adversary can only trace the traffic flow by one hop for one packet transmission.
6. The adversary has enough memory space to save the trace information, and if it receives no more packets for some period
of time, it may backtrack to a previously visited location and then trace the target again.
7. According to Kerckhoff’s Principle [30], the adversary is aware of the routing strategies used in the network.

An adversary initially moves randomly and stays at some place to monitor packets nearby. As soon as it eavesdrops on
a packet, it can determine the location of the immediate sender and move to the location. The adversary can wait there for
the next packet to localize the next sender, by which it can eventually localize the source. To trace the sink, the adversary
needs to identify the transmission direction of the packet and then moves to the current receiver of the packet. Take the
scenario in Fig. 2 as an example, the adversary stays near to node B. To trace the source, the adversary can move to node
A as soon as node A transmits a packet to node B. On the other hand, to trace the sink, the adversary can first identify the
transmission direction of the packet as follows: it detects a packet transmitted from node A when node A sends a packet
to node B; shortly after that, node B transmits a packet and soon again node C transmits another packet; the adversary can
identify the transmission sequence A→B→C and node C is the last receiver. The adversary then moves to node C and waits there to eavesdrop on a next packet to further trace the destination of the traffic flow.

In this paper we will consider both the two adversary models: the patient adversary model and cautious adversary model [8]. In the patient adversary model, the adversary will use the hop-by-hop tracing technique patiently until it captures its target, i.e., the source or sink. While in the cautious adversary model, the adversary will backtrack to its previous location if it waits for a given period of time at certain location without further eavesdropping on any packet. We define the path that the adversary has visited as \( V = \{v_1, v_2, \ldots, v_{l-1}, v_l\} \), where \( v_l \) is the current location of the adversary. When the adversary has not eavesdropped on any new packet within a specific period of time at \( v_l \), it will backtrack along \( V \) to \( v_{l-1} \), and delete \( v_l \) from \( V \) and then wait there for eavesdropping on a new packet. We define the set of locations that the adversary has visited and backtracked as \( F \). To avoid invalid tracing, when the adversary backtracks from \( v_l \) to \( v_{l-1} \), it will add \( v_l \) into \( F \) and ignore packets coming from any location in \( F \). Also, the adversary can avoid getting lost in a loop with loop detection techniques.

### 3.3. Metrics of location privacy protection

We use the following three metrics, safety period, latency, and energy consumption to evaluate the proposed end-to-end location privacy protection schemes:

- **Safety period**: The safety period begins from the moment the adversary initiates the tracing procedure (i.e., eavesdrops on the first packet) and ends at the moment when the adversary captures the source or sink. It is measured in terms of the number of interval length \( T \) before the adversary captures the source or sink.

- **End-to-end latency**: The end-to-end latency is the average time for a packet to travel from source to sink. For simplicity, it is measured in terms of a packet’s average hop count from source to sink.

- **Energy consumption**: As a network consumes much more energy on communication tasks than that on computation tasks, in this paper we only consider the communication cost. We assume that each packet transmission requires an equal amount of energy and the energy consumption is measured in terms of the average number of packets transmitted in the network within period \( T \).

As we have stated, the panda may appear at certain time and will not stay at one place for a long time. Furthermore, the panda may disappear due to its movement into the cave after some period of time. Therefore, the location privacy protection can be considered to be successful if the safety period is longer than the period from the time the panda appears to the time it disappears.

### 4. Location privacy protection schemes against local eavesdropper

In the monitoring system, there is no threat to the location privacy of the source or sink if the panda is not detected, since there is no packet transmitted in the network. However, when a sensor node detects the appearance of the panda, it will become a source node and periodically send the monitoring packets to the sink, which makes the location privacy of the source and sink vulnerable since the adversary can trace either of them by monitoring the transmission flow of the packets. Therefore, the primary purpose of the routing protocols is to protect the location privacy of both the source and sink during the delivery of monitoring packets. This section describes our proposed four end-to-end location privacy protection schemes called forward random walk, bidirectional tree, dynamic bidirectional tree and zigzag bidirectional tree schemes to deliver messages from source to sink.

#### 4.1. Forward random walk scheme (FRW)

In the habitat monitoring system as shown in Fig. 1, the source periodically sends monitoring packets to the sink by multi-hop wireless communications. If the monitoring packets are always delivered from source to sink along a fixed path, it will be easy for an adversary to identify the location of either the source or sink via hop-by-hop tracing. Therefore, a solution to achieve end-to-end location privacy is to randomize the delivery path, based on which we firstly propose the forward random walk scheme (FRW).

The FRW scheme requires each node to obtain its hop count to the sink, which can be achieved by using a sink-based flooding. At the beginning, the sink will initiate a flooding, after which each node can get both its own and its neighbors’ hop counts to the sink. We denote node \( i \)'s hop count to the sink as \( H_i \). Then it satisfies \( |H_i - H_j| \leq 1 \) if nodes \( i \) and \( j \) are the neighbor of each other. The neighbor set of node \( i \) is denoted as \( N_i \).

In the FRW scheme, every node \( i \) divides its neighbors into closer list, equivalent list and further list, which are denoted as \( C_i \), \( E_i \) and \( F_i \) respectively. For node \( i \), each node in \( N_i \) with a hop count smaller than \( H_i \) will be included into \( C_i \), and each node in \( N_i \) with a hop count equal to \( H_i \) will be included into \( E_i \), and each node in \( N_i \) with a hop count larger than \( H_i \) will be included into \( F_i \). We define the forward list of node \( i \) as the union of \( C_i \) and \( E_i \), which is denoted as \( FRL_i \). When a node detects the appearance of the panda, it will be a source node and periodically send monitoring packets to the sink. To forward the monitoring packet, a node will randomly select a neighbor from its forward list as the next hop. Fig. 3 shows one of the message delivery paths of the FRW scheme. Note that the further list will not be considered as the candidates for

---

Please cite this article in press as: H. Chen, W. Lou, On protecting end-to-end location privacy against local eavesdropper in Wireless Sensor Networks, Pervasive and Mobile Computing (2014), http://dx.doi.org/10.1016/j.pmcj.2014.01.006
Algorithm 1 Forward Random Walk Scheme (Node \(i\))

1: Initiation: \(\text{Next}_\text{hop} = \text{null}\).
2: Build the forward list \(\text{FRL}_i\).
3: \textbf{while} receive a message \(m\) \textbf{do}
4: \hspace{1em} Randomly select a neighbor from \(\text{FRL}_i\) as \(\text{Next}_\text{hop}\).
5: \hspace{1em} Forward the received packet to \(\text{Next}_\text{hop}\).
6: \textbf{end while}

Fig. 3. The scenario for the forward random walk scheme.

the next hop since it will route the packet further away from the sink, with a result of remarkably increasing the latency. Consequently, the packet will be transmitted along a forward random walk from source to sink. Algorithm 1 illustrates the procedure of the FRW scheme.

In the FRW scheme, a forward random path is employed, which makes it difficult for the adversary to follow the packets’ delivery path to capture the source or sink. Considering that a packet is currently held by node \(i\) whose hop count to the sink is \(H_i\), the expected number of hops for this packet to be delivered to the sink, denoted as \(x_{H_i}\), can be calculated by the following equation:

\[
x_{H_i} = 1 + x_{H_i-1} \lambda_{H_i} + x_{H_i} (1 - \lambda_{H_i}),
\]

where \(\lambda_{H_i}\) presents the probability that the packet is forwarded from a node whose hop count to the sink is \(H_i\) to a node in its closer list, i.e., \(\lambda_{H_i} = \frac{|CL_i|}{|FRL_i|}\), where \(|CL_i|\) and \(|FRL_i|\) denote the sizes of the lists \(CL_i\) and \(FRL_i\), respectively. From Eq. (1), we can further get:

\[
x_{H_i} = x_{H_i-1} + \frac{1}{\lambda_{H_i}}.
\]

Suppose the source is \(H_s\) hops away from the sink, the expected length of the delivery path is (in terms of number of hop counts):

\[
x_{H_s} = x_{H_s-1} + \frac{1}{\lambda_{H_s}} = x_{H_s-2} + \frac{1}{\lambda_{H_s-1}} + \frac{1}{\lambda_{H_s}}
\]

\[
= x_0 + \frac{1}{\lambda_1} + \cdots + \frac{1}{\lambda_{H_s-1}} + \frac{1}{\lambda_{H_s}}.
\]

It is obvious that \(x_0 = 0\), thus,

\[
x_{H_s} = \sum_{i=1}^{H_s} \frac{1}{\lambda_i}.
\]

Therefore, the end-to-end latency of the FRW scheme is \(\sum_{i=1}^{H_s} \frac{1}{\lambda_i}\) and the energy consumption is \(\sum_{i=1}^{H_s} \frac{1}{\lambda_i}\).

The FRW scheme protects the end-to-end location privacy by randomizing the delivery path. However, it will increase the end-to-end latency since the forward random walk prolongs the delivery path. Furthermore, the FRW scheme relays the packets only to the neighbors in the forward list, resulting in that the safety period cannot be very high. A method to improve the performance is to inject dummy messages into the network. We define the real messages as the monitoring packets transmitted from source to sink and the dummy messages are the packets generated with no useful content but for drawing the adversary away from the actual delivery path.
4.2. Bidirectional tree scheme (BT)

In the hostile WSNs, as the adversary can threaten the location privacy of the source or sink by monitoring the packets flow, a direct way to defend against this threat is to hide the source and sink in the branches of a tree topology formed by the flow of transmitted packets, which makes it much difficult for the adversary to discover them. Therefore, we employ the tree topology in the BT scheme to protect the end-to-end location privacy. Fig. 4 shows the main idea of the BT scheme. The real messages are delivered along the shortest path from source to sink. To protect the source location privacy, topological branches are designed along the shortest path at the source side, in which the dummy messages are delivered from the leaf nodes to the stalk nodes. As the adversary would trace the source by moving backward the direction of the packets, the branches will deviate the adversary from the real delivery path, which can protect the source location privacy. Similarly, the topological branches along the shortest path at the sink side are designed to protect the sink location privacy. The dummy messages in the branches are from the stalk nodes to the leaf nodes, which can draw the adversary away from the real delivery path to protect the sink location privacy since the adversary would trace the sink by moving forward the direction of the packets.

Algorithm 2 Bidirectional Tree Scheme (Node i)

1: Initiation: Next_hop = Null, Child_node = Null.
2: Build the neighbor set $N_i$ and the closer list $C_i$. Randomly select a node from $C_i$ as Next_hop.
3: Child_node ← RandomSelect($N_i$ − Next_hop).
4: while receive a real message m do
5:   Forward the packet to Next_hop.
6:   if $H_i > (1 - \alpha^2)H_s$ then
7:     SetTTL(branch_req, L).
8:     Send branch_req to Child_node with probability P.
9:   else if $H_i < \frac{\alpha^2}{3}H_s$ then
10:    SetTTL(sink_dummy, L).
11:    Send sink_dummy to Child_node with probability P.
12: end if
13: end while

Initially, the sink originates a flooding such that each node can obtain the hop count to the sink. Before sending monitoring packets to the sink, the source delivers a routing request message, which includes its hop count $H_i$, to the sink along the shortest path. For each node $i$ which receives the routing request message, if $H_i > (1 - \alpha^2)H_s$, it will randomly select a neighbor with probability $P$ to generate a branch to protect the source location privacy, where $\alpha$ presents the percentage of nodes in the shortest path to generate the tree branches. Meanwhile, if $H_i < \frac{\alpha^2}{3}H_s$, it will randomly select a neighbor with probability $P$ to generate a branch to protect the sink location privacy. For example, assume $\alpha = 2/3$, then a node $i$ in the shortest path with $H_i > \frac{2}{3}H_s$ will originate a branch with probability $P$ to protect the source location privacy. A node $i$ in the shortest path with $H_i < \frac{1}{3}H_s$ will also originate a branch with probability $P$ to protect the sink location privacy. The nodes in-between just relay the routing request message along the shortest path to the sink.

Algorithm 2 shows the procedure of the BT scheme. When node $i$ receives a real message from a neighbor, it relays the message to a node in $C_i$. In addition, if $H_i > (1 - \frac{\alpha^2}{3})H_s$, node $i$ generates a source side's branch with probability $P$, where the length of the generated branch is $L$. Otherwise, if $H_i < \frac{\alpha^2}{3}H_s$, node $i$ generates a sink side's branch with probability $P$ and length $L$. The generation procedures are described in Algorithms 3 and 4 respectively.
Algorithm 3 Source Side Branch Generation (Node i)
1: Initiation: Child_node = Null, Parent_node = Null.
2: while receive a branch_req message do
3: Set Parent_node as the sender of branch_req.
4: TTL ← GetTTL(branch_req).
5: if TTL > 0 and Child_node = Null then
6: Child_node ← RandomSelect(N_i)
7: SetTTL(branch_req, TTL − 1).
8: Forward branch_req to Child_node.
9: else if TTL = 0 then
10: SetTTL(source_dummy, L).
11: Become a fake source and periodically send source_dummy to Parent_node.
12: end if
13: end while
14: while receive a source_dummy message do
15: TTL ← GetTTL(source_dummy).
16: if TTL > 0 then
17: SetTTL(source_dummy, TTL − 1).
18: Forward source_dummy to Parent_node.
19: end if
20: end while

Algorithm 4 Sink Side Branch Generation (Node i)
1: Initiation: Child_node = Null.
2: while receive a sink_dummy message do
3: TTL ← GetTTL(sink_dummy).
4: if TTL > 0 then
5: if Child_node = Null then
6: Child_node ← RandomSelect(N_i)
7: end if
8: SetTTL(sink_dummy, TTL − 1).
9: Forward sink_dummy to Child_node.
10: end if
11: end while

The dummy messages in the branches can deviate the adversary away from the real delivery path. Thus, the BT scheme can obtain a long safety period against the local eavesdropper. We can also adjust the parameters $\alpha, P$ and $L$ to get a satisfied performance. Assume that the source is $H_s$ hops away from the sink. As the real messages are delivered along the shortest path, the latency will be $H_s$, indicating that the BT scheme can send the monitoring reports to the sink with the minimum end-to-end latency. For the energy consumption, within $T_s$, the average number of transmitted real messages in the network is $H_s$, and the average number of transmitted dummy messages is $\alpha H_s P L$. Thus the energy consumption of the BT scheme is:

$$E_s = H_s + \alpha H_s P L = (1 + \alpha P L) \cdot H_s.$$  \hspace{1cm} (4)

Although the BT scheme can prevent the local eavesdropper from breaking the end-to-end location privacy, there is a potential threat if the eavesdropper can adopt a smarter strategy. As shown in Fig. 4, the adversary may be misled, getting lost in the path between A and the source or the path between B and the sink. However, a smarter adversary may be able to infer the direction of the target based on its visited path $V$. If the adversary is searching for the source when it is near to B, as the real messages are delivered along the shortest path, the adversary can trace hop-by-hop from B to A. Then the adversary can infer that the source should be on the extending line of BA. Then, the adversary can move directly along the direction of BA from A and with a high probability it can identify the source as long as it gets close enough. The adversary can use the similar strategy to infer the direction of the sink as well.

4.3. Dynamic bidirectional tree scheme (DBT)

To prevent the adversary from inferring the direction of the source or sink using the above method, the dynamic bidirectional tree (DBT) scheme combines the FRW scheme and the BT scheme together. Fig. 5 shows the main idea of the DBT scheme. The delivery paths of the real messages vary over time, which can increase the tracing difficulty for the adversary.

Initially, the sink originates a flooding such that each node can get its hop count to the sink. When the source periodically sends monitoring packets to the sink, each node which receives the monitoring packet will randomly select a neighbor from
Fig. 5. The scenario for the dynamic bidirectional tree scheme. The black arrows present the transmissions of real messages and the red arrows present the transmissions of dummy messages. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

its forward list to forward the packet. Therefore, the real messages will travel along a forward random walk from source to sink.

To protect the source location privacy, a dynamic tree topology will be adopted at the source side. Assume that the hop count of the source is $H_s$. When a node $i$ receives a real message from its neighbor $j$, it will forward the real message to the next hop, which is randomly selected from its $FRL_i$. Also, if $H_i/2 < H_i < H_j$, it will generate a source side’s branch with probability $P$ using a method similar to Algorithm 3. The main difference is that each fake source will only send dummy messages for $D$ times. On the other hand, if $H_i < H_i/2$ and $H_i < H_j$, it will generate a sink side’s branch with probability $P$ using a method similar to Algorithm 4. The main difference is that when a node receives a dummy message, it will reselect a child node to relay this dummy message, i.e., the branches at the sink side are dynamic. Algorithm 5 shows the procedure of the DBT scheme.

Algorithm 5 Dynamic Bidirectional Tree Scheme (Node $i$)

1: Initialization: $Next\_hop = null$, $Child\_node = null$.
2: Build the forward list $FRL_i$.
3: while receive a real message $m$ from node $j$ do
4: Randomly select a node from $FRL_i$ as $Next\_hop$ and forward the message to $Next\_hop$.
5: $Child\_node \leftarrow RandomSelect(A_i - Next\_hop)$
6: if $H_s < H_i < H_j$ then
7: SetTTL(branch$_\_req$, $L$).
8: Send branch$_\_req$ to $Child\_node$ with probability $P$.
9: else if $H_i < H_i$ and $H_i < H_j$ then
10: SetTTL(sink$_\_dummy$, $L$).
11: Send sink$_\_dummy$ to $Child\_node$ with probability $P$.
12: end if
13: end while

As the real messages are delivered to the sink along a forward random walk path as described in the FRW scheme, the latency of the DBT scheme is $\sum_{i=1}^{H_s} \frac{1}{\lambda_i}$, where $H_s$ is the hop count of the source to the sink, and $\lambda_i = \frac{|CL_i|}{|FRL_i|}$, and $H_i = i$.

Similar to the FRW scheme, the average number of transmitted real messages within $T_3$ is $\sum_{i=1}^{H_s} \frac{1}{\lambda_i}$. For the average number of transmitted dummy messages, as each fake source will send the dummy messages for $D$ times, the average number of transmitted dummy messages at the source side is $\frac{1}{2} H_i PLD$, where $P$ is the probability for a node to generate a branch and $L$ is the length of each branch. The average number of transmitted dummy messages at the sink side is $\frac{1}{2} H_i PL$. Thus, the energy consumption of the DBT scheme is:

$$\sum_{i=1}^{H_s} \frac{1}{\lambda_i} + \frac{1}{2} H_i PLD + \frac{1}{2} H_i PL = \sum_{i=1}^{H_s} \frac{1}{\lambda_i} + \frac{1}{2} H_i PL(D + 1).$$ (5)

4.4. Zigzag bidirectional tree scheme (ZBT)

The zigzag bidirectional tree scheme (ZBT) is another end-to-end location privacy protection scheme we propose to prevent the adversary from inferring the direction of the source or sink. In the ZBT, we employ the proxy source and the proxy sink to make the real messages be delivered along a zigzag path, which includes three segments: from the source to

Please cite this article in press as: H. Chen, W. Lou, On protecting end-to-end location privacy against local eavesdropper in Wireless Sensor Networks, Pervasive and Mobile Computing (2014), http://dx.doi.org/10.1016/j.pmcj.2014.01.006
the proxy source, from the proxy source to the proxy sink and from the proxy sink to the sink. As shown in Fig. 6, concentric circles A and B represent a proxy source and a proxy sink, respectively. In the path from the source to A, some topological branches will be generated to deviate the adversary away from the delivery path of the real messages to protect the source location privacy. In the second segment, the packets will be delivered along the shortest path from A to B. In the path from B to the sink, some topological branches will also be generated to protect the sink location privacy.

To guarantee the effectiveness of the ZBT scheme, several proxy sink candidates are generated, which are deployed uniformly around the sink. This is to avoid a special unsafe scenario that may exist if only one proxy sink candidate is generated. In this scenario, if the source unfortunately locates very close to this proxy sink, then the functionality of branches at the source side will not work effectively. As shown in Fig. 7, if the source locates close to proxy sink B which is selected as the proxy sink, the path from the source to the proxy source and the path from the proxy source to the proxy sink B will be very close. This will make the source much vulnerable as the adversary will trace from the proxy sink B to capture the source. Note that the number of proxy sink candidates should be carefully determined. If a large number of proxy sink candidates are generated, the proxy source will have more choices to deliver the monitoring packets to the destination, which benefits the location privacy protection. However, since each proxy sink node has to conduct a flooding procedure for each node in the network to get the hop count which is energy-consuming, more cost will be introduced if we use more proxy sink candidates. Thus a trade-off should be made between the number of proxy sink nodes and the induced cost.

In this paper, we use two proxy sink candidates, which are deployed at the two opposite sides of the sink. We make the hop count from the proxy sink to the sink approximate to \(h\), which can be achieved by using one of the following two methods: (1) We can first estimate the average distance per hop in the network denoted as \(d_{\text{hop}}\), and then manually select two nodes, which are \(h \times d_{\text{hop}}\) away from the sink, as the proxy sink candidates. Since the node density in our paper is high (in average, a node has 8.7 neighbors), we consider that \(d_{\text{hop}} \approx R\), thus we can make the distance between the proxy sink and the sink approximate to \(hr\). Also, the selected proxy sink candidates should be on the opposite sides of the sink. (2) The sink can require the nodes, which are \(h\) hops away from the sink, to send a reply message to it, and then select one of the repliers as the proxy sink candidate. After that, the selected proxy sink candidate conducts a flooding procedure to let each node get the hop count to the proxy sink candidate. Then, the nodes which are \(h\) hops away from the sink and \(2h\) hops away from the selected proxy sink candidate will send a request message to the sink. Finally, the sink can select one of them as the other proxy sink candidate.

The sink and the two proxy sink candidates all conduct flooding operations so that each of them can obtain the hop counts to the other two. As the zigzag routing will not work effectively if the proxy sink is close to the source, the ZBT scheme will always select the candidate which is further to the sink as the proxy sink (Proxy sink A in Fig. 7).

To determine the source proxy, the source can generate a \(h\)-hop flooding. Before delivering the monitoring packets to the sink, the source will select a node \(h\) hops away from itself as the source proxy. Note that the proxy source should be carefully selected to make the path from the source to the proxy source away from the sink, making the sink safe enough when the adversary traces along this path.

Similar to the BT scheme, as shown in Fig. 6, when the monitoring packet is delivered from the source to the proxy source, each node in the path will generate a branch with probability \(P\) and length \(L\). Similarly, when the monitoring packet is delivered from the proxy sink to the sink, each node in the path will also generate a branch with probability \(P\) and length \(L\). The monitoring message will be delivered along the shortest path from the proxy source to the proxy sink without generating any branch. Algorithm 6 describes the procedure of ZBT scheme, and the branch generation procedures at the source and sink sides can be referred to Algorithms 3 and 4 respectively.

We denote the hop count from the proxy source to proxy sink as \(H_p\), then the end-to-end latency of the ZBT scheme is \(2h + H_p\), where \(h\) indicates the hop count from the proxy source to source and from the proxy sink to sink. For the energy consumption, the average number of transmitted real messages within interval \(T_S\) is \(2h + H_p\), and the average number of
Algorithm 6 Zigzag Bidirectional Tree Scheme (Node $i$)

1: Initiation: $\text{Next}_\text{hop} = \text{null}$.
2: \textbf{while} receive a real message $m$ \textbf{do}
3: \hspace{1em} $\text{Destination} \leftarrow \text{GetDestination}(m)$.
4: \hspace{1em} \textbf{if} $\text{IsProxySource}(\text{Destination}) = \text{true}$ \textbf{then}
5: \hspace{2em} \textbf{if} $\text{IsProxySource}(i) = \text{true}$ \textbf{then}
6: \hspace{3em} Determine $\text{Next}_\text{hop}$ and forward $m$ towards proxy sink.
7: \hspace{2em} \textbf{else}
8: \hspace{3em} Determine $\text{Next}_\text{hop}$ and forward $m$ towards proxy source.
9: \hspace{2em} SetTTL($\text{branch}_\text{req}$, $L$).
10: \hspace{2em} Send $\text{branch}_\text{req}$ to $\text{Child}_\text{node}$ with probability $P$.
11: \hspace{1em} \textbf{end if}
12: \hspace{1em} \textbf{else if} $\text{IsProxySink}(\text{destination}) = \text{true}$ \textbf{then}
13: \hspace{2em} \textbf{if} $\text{IsProxySink}(i) = \text{true}$ \textbf{then}
14: \hspace{3em} Determine $\text{Next}_\text{hop}$ and forward $m$ towards sink.
15: \hspace{2em} \textbf{else}
16: \hspace{3em} Determine $\text{Next}_\text{hop}$ and forward $m$ towards proxy sink.
17: \hspace{2em} \textbf{end if}
18: \hspace{1em} \textbf{else if} $\text{IsSink}(\text{destination}) = \text{true}$ \textbf{then}
19: \hspace{2em} \textbf{if} $\text{IsSink}(i) = \text{false}$ \textbf{then}
20: \hspace{3em} Determine $\text{Next}_\text{hop}$ and forward $m$ towards sink.
21: \hspace{3em} $\text{Child}_\text{node} \leftarrow \text{RandomSelect}(\mathcal{N}_i - \text{Next}_\text{hop})$.
22: \hspace{3em} SetTTL($\text{sink}_\text{dummy}$, $L$).
23: \hspace{3em} Send $\text{sink}_\text{dummy}$ to $\text{Child}_\text{node}$ with probability $P$.
24: \hspace{2em} \textbf{end if}
25: \hspace{2em} \textbf{end if}
26: \hspace{1em} \textbf{end if}
27: \textbf{end while}

transmitted dummy messages is $2hPL$. Thus, the energy consumption of the ZBT scheme is:

$$2h + H_p + 2hPL = 2PL + 1 + H_p.$$  \hspace{1em} (6)

Note that $2h + H_p > H_s$ where $H_s$ is the source node's hop count to the sink, which indicates that the energy consumption of the ZBT scheme is larger than $H_s$, i.e., $2PL + 1 + H_p > H_s$.

5. Performance evaluation

We implement our proposed end-to-end location privacy protection schemes on the TOSSIM [31] platform to illustrate their effectiveness. We evaluate our proposed schemes based on the previously mentioned metrics, i.e., safety period, latency and energy consumption. Two different adversary models, i.e., the patient adversary model and the cautious adversary model are considered in the simulations. We compare the proposed four location privacy protection schemes with the baseline—the shortest path scheme, in which the message will be delivered from source to sink along the shortest path.

5.1. Simulation settings

The simulation parameters are set as follows: The network is generated by uniformly deploying 3000 sensor nodes within a rectangular area of $9000 \times 2700$ m$^2$. The communication range $R$ of each sensor node is 150 m. The average number of
neighbors for a node is 8.76. Each node that relays the real message has the probability $P = 0.8$ to generate a branch with a length $L = 10$. For the BT scheme, $\alpha$ is set as 1. For the ZBT scheme, the hop count from the proxy source to the source and from the proxy sink to the sink is $h = 15$. For the DBT scheme, $D = 5$. And the hop count from the source to the sink, $H_s$, varies from 10 to 35 with an increment of 5.

5.2. Simulation results

Fig. 8 shows the safety period of the source location privacy of these five schemes (our proposed four schemes plus the shortest path scheme) under the patient adversary model. It is obvious that the ZBT scheme achieves the highest safety period. The safety period of the BT scheme increases rapidly as the hop count increases. When the hop count is larger than 30, the safety period of the BT scheme gets close to that of the ZBT scheme. The safety period of the FRW and DBT schemes are relatively low, and the DBT scheme slightly outperforms the FRW scheme. Obviously, all the four schemes outperform the shortest path scheme.

The safety period of the sink location privacy of these five schemes under the patient adversary model is shown in Fig. 9. As the adversary needs to wait for the packet relaying to determine the transmission direction of the packet before it can move to the receiver, the safety period of the sink location privacy is larger than that of the source location privacy. We can find that the ZBT scheme outperforms other schemes. When the hop count is larger than 15, the safety periods of the ZBT, BT and FRW schemes tend to be larger than 200, indicating a high sink location privacy. Figs. 8 and 9 show that the proposed four schemes all outperform the shortest path scheme. Both figures also illustrate that the DBT scheme cannot achieve a high location privacy under the patient adversary model.

Under the cautious adversary model, the safety period of the source location privacy of these five schemes is shown in Fig. 10. The safety period of the source location privacy under the cautious adversary is lower than that under the patient adversary.
adversary model. This is due to the reason that the cautious adversary is smarter: when it waits for a long time at a certain location, it can trace back to previous locations to avoid being deviated away by some dummy messages. When the hop count equals to 10, the ZBT scheme obtains the highest safety period. However, the DBT scheme outperforms other schemes when the hop count is larger than 15. The safety period of all the schemes increases with the increase of hop count. However, the increase ratio of the ZBT scheme is relative small. When the hop is larger than 30, the FRW scheme outperforms the ZBT scheme. The performance of the BT scheme is always worse than the FRW, DBT and ZBT schemes. All the proposed four schemes outperform the shortest path scheme.

Fig. 11 illustrates the safety period of the sink location privacy of these five schemes under the cautious adversary model. Compared to the sink location privacy under the patient adversary model, the safety period is lower under the cautious adversary model as the adversary is more capable in tracing the packet transmission. The FRW scheme achieves the highest performance while the shortest path scheme has the lowest safety period. As it is more time-consuming for the adversary to capture the sink than the source, the safety period of the sink location privacy is also larger than that of the source location privacy.

Fig. 12 shows the end-to-end latency of these schemes. For the BT scheme and the shortest path scheme, as the real messages are delivered along the shortest path from source to sink, they would achieve the shortest end-to-end latency. Since the real messages in the FRW and DBT schemes are delivered along the forward random walk path, the end-to-end latency of these two schemes is similar. When the hop count equals to 10, the end-to-end latency of the ZBT scheme is the largest as it employs the zigzag path to mislead the adversary. When the hop count is larger than 15, the end-to-end latency of the FRW and DBT exceeds that of the ZBT scheme.
Fig. 13 shows the energy consumption of these schemes. The shortest path scheme consumes the least energy since it does not generate any dummy message and the real messages are delivered along the shortest path. The FRW scheme consumes the second least energy due to the reason that it does not generate any dummy message. When the hop count is less than 20, the ZBT scheme consumes the most energy because a further proxy sink would be selected and more dummy messages are generated than the other schemes. When the hop count is larger than 20, the energy consumption of the DBT scheme is the largest, because more branches are employed than other schemes.

6. Conclusion and future work

The end-to-end location privacy is an important issue in WSNs. In this paper, we address the necessity of simultaneously protecting the location privacy of both the source and sink in the habitat monitoring system. We propose four location privacy protection schemes, forward random walk, bidirectional tree, dynamic bidirectional tree and zigzag bidirectional tree, to deliver messages from source to sink, which can protect the end-to-end location privacy against the local eavesdropper. We also implement the proposed schemes on the TOSSIM platform, and evaluate the performance in terms of safety period, end-to-end latency and energy consumption. The simulation results illustrate that our proposed location privacy protection schemes can obtain satisfied performance.

Since each of our proposed schemes has different performance on protecting the source location privacy or sink location privacy, as the future work, we plan to decompose our proposed schemes and analyze the location privacy protection at the source and sink respectively. We will then design an optimal combination from the decomposed schemes to achieve a highest location privacy protection for both ends. As we only consider a single stationary source in this paper, another direction of our future work will focus on extending our schemes to be applicable to multiple mobile sources.
Acknowledgments

This work was supported in part by NSFC grants (No. 61309023, No. 61272463), Shandong Provincial Natural Science Foundation, China (No. ZR2013FQ032), the Fundamental Research Funds for the Central Universities (No. 13CX02100A), Open Project in Zhejiang Provincial Key Lab of Intelligent Processing Research of Visual Media (No. 2012008), Hong Kong GRF grants (PolyU-524308, PolyU-521312), HKPU grants (A-PL16, A-PL84).

References


