On Providing Location Privacy for Mobile Sinks in Wireless Sensor Networks

Edith C.-H. Ngai and Ioana Rodhe
Department of Information Technology
Uppsala University, Uppsala, Sweden
{edith.ngai, ioana.rodhe}@it.uu.se

ABSTRACT

Wireless sensor networks have attracted increasing attentions considering their potentials for being widely adopted in both emerging civil and military applications. A common practice of sensor networks is to collect data from the sensors and report the data to the sinks or to some pre-defined data rendezvous points via multi-hop communications. Attackers may locate a sink easily by reading the destination field in the packet header or predicting the arrival of a sink at the rendezvous points, which opens up vulnerabilities to location privacy of the sinks. In this work, we propose a random data collection scheme to protect the location privacy of mobile sinks in sensor networks. Data are forwarded along some random paths and stored at random nodes in the network. The sinks move around along some random paths to collect data from the local nodes occasionally, which prevents the attackers from predicting their locations and movements. We analyze different attacks threatening the location privacy of the sinks in sensor networks. We also evaluate the delivery rate, data collection delay and protection strength of our scheme by both analysis and simulations. The results show that our scheme can provide location privacy of mobile sinks effectively, while providing satisfactory data collection services.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: General; C.4 [**Performance of Systems**]: Reliability, availability, and serviceability; D.4.6 [**Security and Protection**]: Information flow controls

General Terms

Security, Performance

Keywords

Wireless sensor networks, Privacy, Data Collection

MSWiM'09, October 26–29, 2009, Tenerife, Canary Islands, Spain.
Copyright 2009 ACM 978-1-60558-616-9/09/10 ...$10.00.

1. INTRODUCTION

A wireless sensor network (WSN) is composed of numerous small sensing devices with limited communication range. The sensors collect data from the environment and report them to the sinks. With the promising sensing and wireless technologies, sensor networks are expected to be widely deployed in a broad spectrum of civil and military applications in the near future [1]. Location information of the sinks, the sensors, and the objects being tracked are very important in sensor networks. Protecting location privacy in sensor networks is crucial considering different kinds of attacks that may disrupt the normal function of the networks.

In many applications, sensors report measurements to the sinks via hop-by-hop communications [14, 15, 24]. Most of the existing routing protocols based on geographic routing require location of the sinks to be known among the sensors [2, 4, 8, 13], which may pose higher dangers to the sinks to become the attack targets. In geographic routing, a sensor usually forwards the packets to the next hop that is closest to the sink, though sometimes it may also consider some additional factors, like delay and energy consumption [4, 8]. In order to route a packet to the sink, a sensor must know the destination field of the packet or the location of the sink. Alternatively, sensors in networks with mobile sinks can also forward and store the data at some rendezvous points [10, 19, 22]. Sensors have to know the location of the sinks or the rendezvous points in this approach. However, this mechanism allows the adversary to locate and attack the sink easily. To address this problem, we propose a data collection scheme that allows data storage and collection along random paths to protect the location privacy of mobile sinks.

Location privacy in sensor networks has attracted much attentions recently. The destination nodes or the sinks, whose locations are discovered by the adversary, may become the target of attacks. For example, a soldier who carries a receiver will be in great danger if being discovered. It is therefore very important to protect sink location privacy in sensor networks. Traffic-analysis attacks, which are performed by an adversary who discovers the receiver location by observing the flow of network traffic, have been widely studied. The problem was addressed by dummy packets injection, but this approach increases the network traffic heavily [11, 12, 23]. In addition, it does not consider active attackers who can compromise a node and read the header field of a packet to identify the receiver.

In this paper, we propose a random data collection scheme for sensor networks with mobile sinks, which can keep the location and movement of the sinks private in the network.

Sensors cannot trace or predict the movement of the sinks, though they can still report the data effectively. Similar to other approaches with mobile sinks [10, 19, 22], the sinks will approach the sensors and collect the stored data. However, instead of forwarding the data to some pre-defined rendezvous points, the sensors forward the data to some random nodes for temporarily storage. Different from moving along periodic paths, the sinks move randomly in the network to collect data. Our scheme can prevent the attackers from predicting the movement of the sinks, but still providing satisfactory data delivery services.

The remainder of the paper is organized as follows. In Section 2, we describe some related work in the area. In Section 3, we discuss the network model and the threat model. In Section 4, we present our random data collection scheme which provides location privacy of mobile sinks in WSNs. Sections 5 and 6 summarize the analytical and simulation results, and we conclude the paper in Section 7.

2. RELATED WORK

Privacy issues in sensor networks, especially location privacy [6, 11, 12], have been studied in recent years. The random walk based phantom flooding scheme [12] is proposed to defend against an external adversary who attempts to trace back to the data source in sensor networks and provide source location privacy. A path perturbation algorithm [9] is also proposed to cross paths in areas where at least two users meet which intends to make the attackers confuse the paths of different users. Although the random routing approach can protect the network from local adversaries who overhear and analyze the traffic passively, it cannot defend against active attackers who are able to capture the packets and read the receiver location in the destination field.

Other schemes, like ConstRate and ProbRate, which introduce dummy traffic to hide the real event sources, are proposed to provide source event unobservability in the network [18, 23]. Even though some dummy packets can be dropped on their way [18], the injected dummy traffic still increases the packet delay and consumes more energy in sensor nodes. Also, these schemes focus on source privacy, which is different from our goal of protecting the location privacy of the sinks.

Multpath routing and fake message injection are introduced in [3] to provide receiver privacy. However, it concentrates on the traffic-analysis attack, which determines the location of the sink through the measurement of traffic rates at various locations. Another recent work is proposed to protect receiver-location privacy in WSNs by providing path diversity in combination with fake packet injection [11]. It is solving a similar problem as we do, but it considers only passive attackers who capture the receiver by eavesdropping and performing network traffic analysis. In our work, we also protect the network from active attackers who can compromise an intermediate node and capture the packets. Our random data collection scheme can keep the location of the sink private to the nodes in the network. It excludes the location of the sink in the header of the packets and prevents the attackers from predicting the movement of the sinks. Moreover, our approach does not require injection of extra fake packets, so the network traffic can be reduced.

There are also some related work on data collection with mobile sinks. Shah et al. [17] modelled the performance of the sink based on the random mobility model. Several heuristics are proposed in [7, 20] to schedule the movement of sink such that the source nodes can be visited before buffer overflow. There are also some work that jointly consider multi-hop network transmissions and the movement of the sink in data collection. The rendezvous approach has been widely studied in which sensors send the data to some selected rendezvous points for temporary storage until the sinks come and collect them. In [10, 19], data are forwarded from the sources to the nodes close to the path of the sink. The sink then picks up the cached data when it passes by. Wang et al. [21] showed that constraining the mobile relays in the vicinity of the sink can maximize the network lifetime. Xing et al. [22] proposed two algorithms for planning the data collection tours of mobile sinks in which the mobile sinks travel along the network routing trees. Different from the above work, we look at the security aspect of this problem, rather than optimizing the performance of packet delay or energy consumption. We aim at providing location privacy of the sinks to protect them from being traced or attacked, while still providing satisfactory data collection services in sensor networks.

3. NETWORK AND THREAT MODELS

3.1 Network Model

We consider a wireless sensor network consisting of a number of sensors deployed in an area, together with one or multiple mobile sink(s). Each sensor has a limited transmission range for wireless communication which allows it to exchange messages directly with its neighboring nodes. Sensors collect data and store them temporarily in the network. The sinks will walk randomly in the field and broadcast occasionally to some local sensors to collect data.

Since sensors have limited storage, communication range and computation power, they cannot afford the relatively heavy-load asymmetric cryptography. Instead, they use symmetric cryptographic primitives to provide data confidentially, authentication, integrity, and freshness of the message [5, 16]. We assume that each sensor i shares an unique symmetric key K_i with the sink. Note that multiple sinks can share the same symmetric key K_i with i.

3.2 Threat Model

We consider attackers who aim at tracing and attacking the sinks. They may discover the location of the sink by reading the destination field of the packets, following the network data flows or predicting the movement of the sink. We summarize the common attacks into three categories as follows.

3.2.1 *Capturing the Packets*

Active attackers may capture a node and read the packets passing through. They can read the destination field of a packet to find out the location of the sink. The widely adopted geographic routing protocols in sensor networks [2, 4, 8, 13] are vulnerable to this kind of attack as the location of the receiver must be included in the destination field of a packet for routing.

3.2.2 *Observing Network Traffic*

Some attackers may monitor the network traffic passively to predict the location of the receiver [3, 11]. Since the receiver is likely to be the sink in many sensor network applications, the attackers may notice a large amount of traffic flowing towards the sink. These passive attackers are usually equipped with some supporting devices, such as an antenna, which allow them to eavesdrop the delivery of packets and perform some simple traffic analysis. They can also predict the direction of the receiver based on the signals that they overheard.

3.2.3 Predicting Movement of Sink

Some attackers may trace the sink by predicting its movement. They may also wait for the arrival of the sink by staying at the same place. The traditional approaches in sensor networks with mobile sinks visiting rendezvous points periodically [10, 19] are particularly vulnerable to this kind of attacks.

3.3 Notations

We use the notations in Table 1 to describe and analyze our random data collection scheme.

Table 1: Notations

N_s	Number of sensors in the network
N_c	Average number of copies on a piece of data
N_b	Average number of one-hop neighbors
p_d	Probability for a sensor to generate new data
p_s	Probability for an intermediate node to store the data
B	Buffer size (in number of data) of a sensor
L	Length of random path to forward data
a	Time interval for sink to collect data
p_a	Probability that a node will get a new piece of data in its buffer in one time unit
q_t	Number of newly arrived data to a node in time t
$P[q_t < B]$	Probability that the stored data is still in buffer after time t
$P[S_i]$	Probability that the sink will visit a node in i visits
$P(S)$	Probability that the packet is delivered to the sink successfully
$E[i]$	Average number of collections before the data are collected by the sink
$E[T]$	Average data collection time
E_t	Energy for transmitting a packet
E_r	Energy for receiving a packet

4. RANDOM DATA COLLECTION SCHEME

We describe our random data collection scheme which protects the location privacy of mobile sinks in WSNs. It is composed of random data forwarding and storage of the sensors and random movement and data collection of the mobile sink.

4.1 Random Data Forwarding and Storage

When a sensor i reports its measurement to the sink, it encrypts the message with its symmetric key K_i and forwards the data along a random path. Unlike many existing routing algorithms [2, 4, 8, 13], the location or ID of the sink is not included in the packet when forwarding the data. The advantage of this approach is to avoid the attackers from obtaining the destination of the packet even though they are overhearing the messages at the intermediate nodes.

When a node generates a piece of data, it will store a copy locally. Then, it will forward the data to some nodes for storage and report to the sink. Since the node does not know the location of the sink, it forwards the packet randomly to any of its neighbors. When the next hop j receives the packet, it again forwards the packet to one of its neighbors randomly and increases the hop count field h in the packet by one. However, it will not forward the packet back to the previous hop. The hop count field h in the header of the packet is initialized to zero by the source node. It indicates the number of hops that the packet has travelled. The above forwarding process repeats hop-by-hop until $h = L$, where L is the pre-defined length of the random path.

When the intermediate nodes receive a packet, each of them has a probability p_s to store the data in its buffer. If the buffer is full, the node will remove the oldest data to free space for the newly arrived data.

Figure 1 shows an example of data forwarding and storage along a random path. The packet from the source node s is forwarded to a number of nodes along a random path. The intermediate nodes store a copy of the data probabilistically. Since the packet does not include any destination field, an adversary $A1$ cannot obtain the location of the receiver even though it can capture an intermediate node and read the packet. In our scheme, all sensors, including the source s, do not know the location of the sink. The packet keeps travelling until the path length $L = 8$. Consider another adversary $A2$, which is equipped with an antenna to overhear the network traffic, he cannot predict the sink location by traffic monitoring as the packet travels along a random path with no specific destination. The flow of the packet in our scheme is totally independent of the location of the sink.

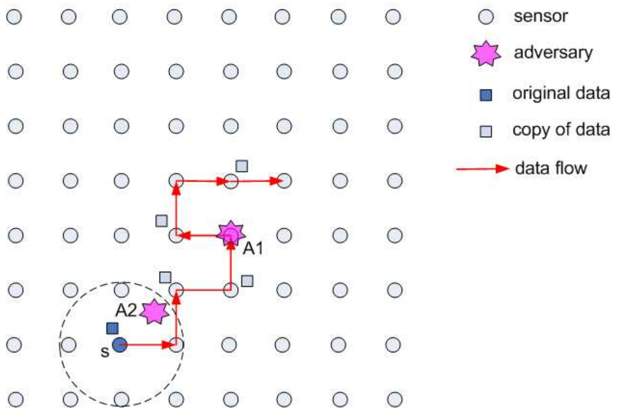

Figure 1: Random data forwarding and storage for protecting sink location privacy in sensor networks. Source node s forwards the packet along a random path with length $L = 8$. Random forwarding includes no destination field in a packet. The scheme can protect the location privacy of the sink against attackers $A1$ and $A2$, who can capture an intermediate node and observe the network traffic.

4.2 Random Movement of Sinks in Data Collection

The mobile sink moves around the network to collect data from the sensors. To avoid being attacked and tracked, it changes its moving direction randomly and only requests data from its local neighbors occasionally. In each broadcast, the sink will collect all the data in the buffer of its neighboring nodes as shown in Figure 2. Then, it will filter out the data that have already been received. Only the data received for the first time will be recorded and reported to the users. The neighboring nodes will free their buffer after reporting all the data to the sink.

Since the sink broadcasts only to a limited number of neighboring nodes, the chance for an attacker to know the sink's current location is low. It will also be quite impossible for the attacker to trace or predict the movement of the mobile sink due to its random movement.

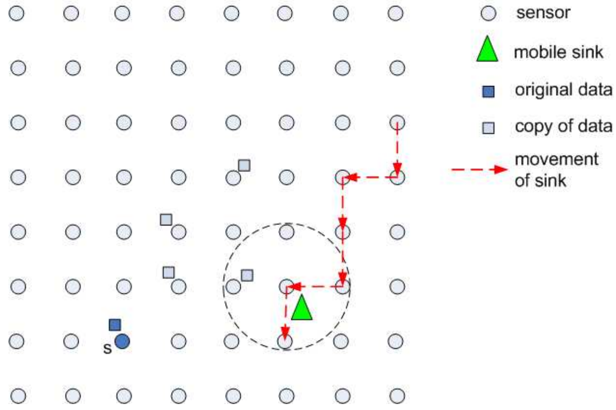

Figure 2: The mobile sink moves around in the network in random directions. It broadcasts to collect data from its local neighbors every 5s. The local nodes then report their stored data to the sink and free their buffers.

5. ANALYTICAL RESULTS

We consider that the sink moves one step to any of its neighboring nodes in each time unit. We assume that the locations of the sink and the data follow uniform distribution, so that the attackers cannot predict their locations based on any pattern.

5.1 Delivery Probability

A packet is delivered successfully if it reaches at least one of the sinks in the network. We denote $P(S)$ as the probability that the packet is delivered to the sink successfully in our scheme.

Let p_d be the probability for a sensor to generate a new data in one time unit, N_s be the total number of nodes in the network and N_c be the average number of copies stored in the network for a piece of data. The average total number of data generated and copied in the network in one time unit is then $N_t = p_s N_s N_c$. Given that N_t is small compared with N_s, a sensor may get either zero or one copy of data in one time unit. The average probability that a node will get a new piece of data to its buffer in one time unit, p_a, can be calculated as

$$p_a = \frac{p_d N_s N_c}{N_s} = p_d((L-1)p_s + 1), \tag{1}$$

where $N_c = (L-1)p_s + 1$, L is the length of the random path and p_s is the probability for an intermediate node to store the data in its buffer.

Suppose that the generation time of a particular piece of data is t_0, a copy of the data will be stored in the source node as well as in some intermediate nodes along the random path. After time t, the probability that the piece of data is still in the buffer of a storage node is equal to the probability that the incoming data from t_0 to t is less than B, where B is the buffer size of a node in number of data.

A packet is delivered successfully if the sink visits at least one of the storage nodes. The probability that the number of incoming packets q_t between t_0 and t is less than B can be calculated by

$$P[q_t < B] = \begin{cases} 1 & \text{if } t < B \\ \sum_{k=0}^{B-1} \binom{t}{k} p_a^k (1-p_a)^{t-k} & \text{else} \end{cases} \tag{2}$$

which indicates the probability that the buffer is not overflowed, so that the piece of data is still in the buffer of the storage node.

We also calculate the probability $P[S_i]$ that the sink will visit a node at the i^{th} visit or at time t, where $t = ai$ and a is the time interval between two collections.

$$P[S_i] = (1-p)^{i-1} p, \tag{3}$$

where p is the probability that the sink can receive data from a storage node. If N_c is small and the data copies are dispersed, p can be obtained by

$$p = \begin{cases} \frac{N_c N_b}{N_s} & \text{if } N_c N_b < N_s \\ 1 & \text{else} \end{cases} \tag{4}$$

where N_b is the number of local nodes that report data to the sink in one data collection.

The successful delivery probability $P(S)$ can then be obtained by

$$P(S) = \sum_{i=1}^{\infty} P[S_i] P[q_{t=ai} < B], \tag{5}$$

where q_t is the amount of data arrived between time t_0 and t, where $t = ai$ corresponding to the i^{th} data collection by the sink.

Figure 3 shows the successful delivery probability of the data varying the buffer size B. The successful delivery probability increases with B as less packets are lost due to buffer overflow. Note that the delivery probability of $L = 20$ is quite low when B is small as the buffers overflow more easily with more copies of data in the network.

5.2 Data Collection Delay

Data collection delay is the time taken for a data to be collected by the sink. Since the data transmission time is insignificant compared with the moving time of the sink, the delay is approximately the time taken for the sink to arrive at the storage nodes and collect the data. If the packet reaches the sink(s) more than once, the delay is measured considering the time that the packet reaches the sink for the first time.

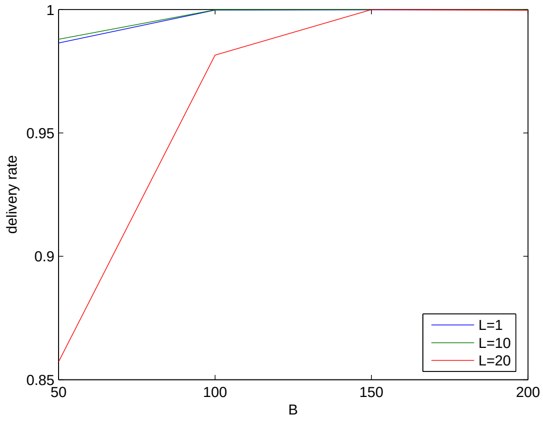

Figure 3: Successful delivery probability varying buffer size B. Both $L = 1$ and $L = 10$ achieve high delivery probability, while $L = 20$ has the lowest delivery probability due to buffer overflows.

The average number of data collections $E[i]$ before the data is collected by the sink can be calculated as

$$E[i] = \sum_{i=1}^{\infty} P[S_i]P[q_{t=ai} < B]i. \qquad (6)$$

The average data collection time will be

$$E[T] = E[i]a. \qquad (7)$$

Figure 4 shows the average delay for data collection varying B. Although the successful delivery probability of $L = 1$ and $L = 10$ are high, they are suffering from long data collection delay. On the other hand, the delay of $L = 20$ is much lower as there are more copies of the data stored in the network, so the mobile sink has a higher chance to collect them earlier. The figure also shows that the data collection delay becomes constant after the buffer size is increased to a certain level.

5.3 Security Analysis

We analyze the protection strength of our scheme against different types of attacks which aim at capturing or tracing the sink.

5.3.1 Capturing the Packets

In our random data collection scheme, the sensors only report data to the sink when the sink arrives at their neighborhood. The communications are limited to the sink's local neighbors. Also, the data are forwarded along random paths for temporary storage. Routing from the sensors to the sink is not required, so the attackers cannot locate the sink by reading the destination field of the packets.

5.3.2 Observing Network Traffic

The sink moves around to collect data only from its local neighbors in our scheme. The sensors report data to the sink only when it approaches them. This mechanism will not create large amounts of traffic from sensors to the the sink, so passive attackers are unable to capture the sink by observing the network flow.

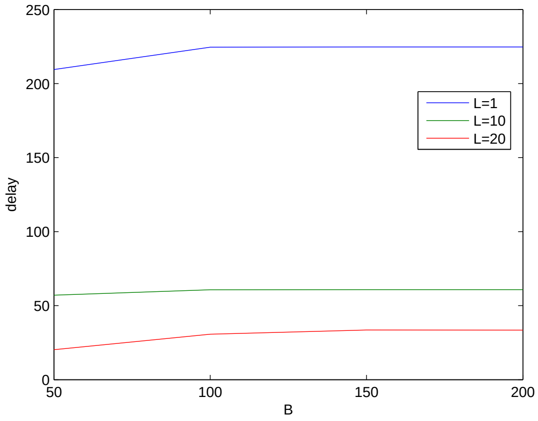

Figure 4: Data collection delay (in seconds) varying buffer size B. $L = 20$ has the lowest delay as there are more copies of data in the network. $L = 1$ has the highest delay as data are stored only at the source nodes.

5.3.3 Predicting Movement of Sink

The sink can move at d different directions randomly to collect data in our scheme. It broadcasts to its local neighbors N_b every time interval a. If an attacker waits at one location, it has a probability N_b/N_s to receive the broadcast from the sink in one broadcast. Suppose that the sink will move for a steps before another broadcast. The probability that the attacker can trace the sink step by step to the next broadcast becomes $p_a = (1/d)^a$. Given that a sink can move at 4 directions and it broadcasts every 5 steps, $p_a = 0.00098$ which is very low. Apart from tracing, the probability for the attacker to meet the sink at the next stop by waiting at the same location is also low due to the random walk of the sink. The attacker will have to wait for a long and unpredictable period of time to meet the sink.

6. SIMULATION RESULTS

We evaluate the performance of our random data collection scheme with simulations. The network considered has a total of 225 sensors and a mobile sink. The nodes are uniformly distributed over a regular grid within a 560m x 560m square. Each node has an equal distance (40 meters) to its neighboring nodes. The wireless communication range is 45 meters, such that each node can only receive signals from its four closest neighbors.

A node generates data with probability p_d every second. The data will be forwarded L hops in the network on a random path. The intermediate nodes which receive the data will store them with a probability p_s in their buffer of size B. If the buffer is full, the oldest data will be removed to make place for the new data.

The mobile sink moves to a new node every second at a random direction. It collects data from its local neighbors every 5s. We run the simulation for 4000s and stop generating data in the last 1000s. The successful delivery rate and the packet delay are measured.

We conduct two series of experiments. The first experiment monitors the change of delivery rate and packet delay along the simulation time. In the second experiment, we measure the delivery rate and packet delay at the end of the simulation and study the impacts of the parameters B and L.

6.1 Transient Behavior

In the first experiment, we run the simulation for 4000s and calculate the delivery rate and the packet delay for the generated packets every 100s. We set $p_s = 0.3$, $B = 150$, $p_d = 0.05$ and plot the results varying L from 1 to 20. The results of the delivery rate are presented in Figure 5 with the corresponding packet delay in Figure 6.

At the beginning of the simulation, the delivery rate is low because the data have just been generated and the sink has not visited and collected most of them yet. In the first 1000s, the delivery rate increases rapidly since copies of the data are sent and stored around in the network and the sink collects them constantly. From 1000s to 3000s, the delivery rate stabilizes at around 0.8 to 0.9 and increases only slightly. As long as data are generated continuously, the sink will need more time to collect all of them. In order to see how much data can be collected in our scheme eventually, we stop generating data after 3000s but continue to collect them for another 1000s. The simulation results show that the sink manages to collect almost all the data at the end of the simulation with the delivery rate close to 1.0.

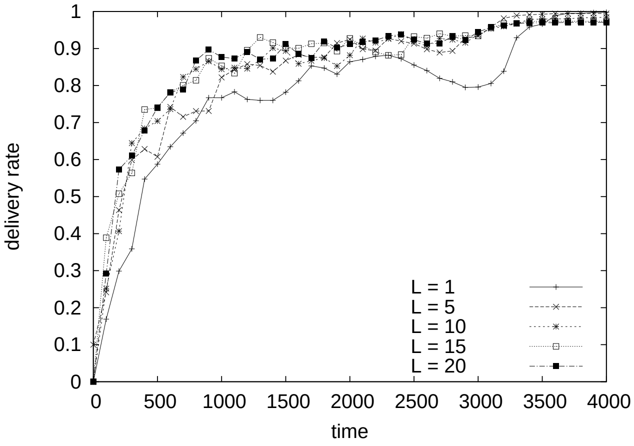

Figure 5: Delivery rate along a 4000s simulation.

The packet delay is the time taken for the data to be collected by the sink. It is measured in unit of seconds and only for the packets that are collected by sink successfully. The packet delay is very low at the beginning of the experiment as the data collected by the sink are all recently generated. In the first 1000s, the packet delay increases gradually since the sink starts to collect both old and new data. After 1000s, when the delivery rate stabilizes, the packet delay also stabilizes. In the last 1000s, the packet delay increases further because all the data collected by the sink at the moment were generated in the first 3000s. Note that increase of packet delay in the last 1000s is most obvious for $L = 1$. In this case, the data are stored only at the source node, so it takes time for the sink to visit all the nodes and collect their data.

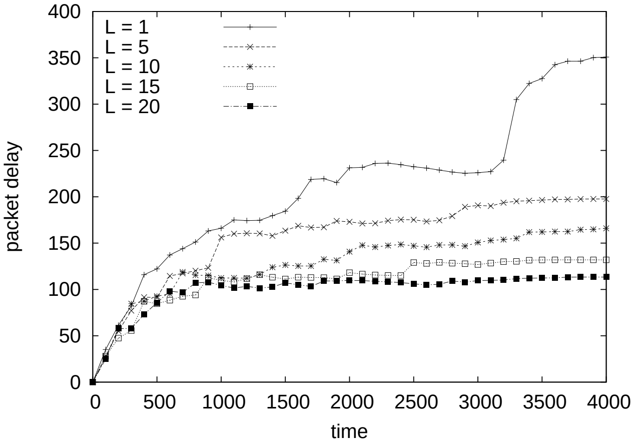

Figure 6: Packet delay (in seconds) along a 4000s simulation.

6.2 Impact of Buffer Size B

In the second experiment, we investigate how the buffer size affects the delivery rate and the packet delay. We vary B from 50 to 200 with $p_s = 0.3$ and $L = 1, 10$ and 20 as shown in Figure 7 and Figure 8. From Figure 7, the delivery rate increases with the buffer size B as the buffers do not overflow so quickly. The data, which can stay in the buffer longer, have higher probability to be collected by the sink eventually. However, the data being stored in the buffer longer also result in higher packet delay as shown in Figure 8. We believe that the packets which increase the packet delay with large B are those being lost due to buffer overflow with small B. Therefore, the increased average packet delay here does not mean that the collection time of all data becomes high. Although the case with $L = 1$ has better delivery rate than $L = 10$ and 20, it has about double packet delay.

The simulation results also match well with our analytical results in Figures 3 and 4. The curves in the analytical results and simulation results share similar shapes, though the delivery rates in the analytical results are slightly higher than those in the simulation results. The reason is that we consider the probability for a node to get a piece of data p_a in each time unit in our analytical model. In simulations, however, a node may receive more than one piece of data in one time unit. The probability for a node to have buffer overflow then becomes higher in the simulations than in the analysis. The number of storage nodes may also be less than the copies of data in simulations, while the data are distributed more evenly in our analysis. Hence, the delivery rate in the analysis is higher than that in the simulations. The same reason explains the shorter data collection delay in Figure 4 as the sink can get the data from more storage nodes.

6.3 Protection Strength

We evaluate the protection strength of our Random Data Collection (RDC) scheme against the three types of attacks discussed before. We evaluate the probability that the attacker can capture the location of the sink by (1) reading the destination field of a packet, (2) observing the network traffic or (3) predicting the next location of the sink. Given

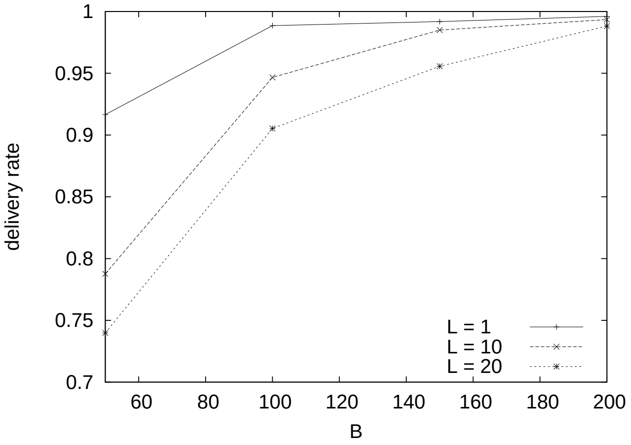

Figure 7: Successful delivery rate varying the buffer size B.

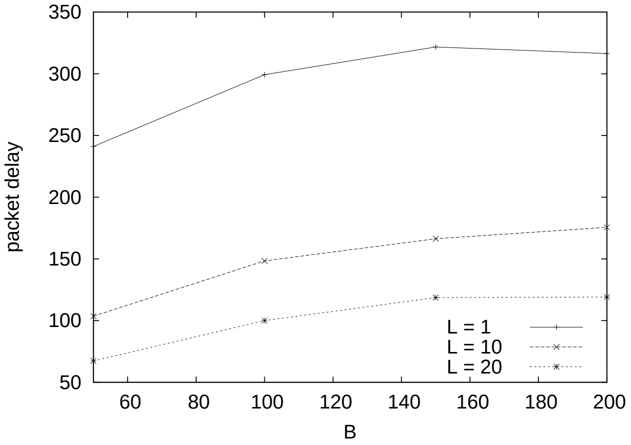

Figure 8: Average packet delay (in seconds) varying the buffer size B.

that some attackers may stay at the same place to wait for the sink, we also measure the time taken for them to meet the sink again. The results are shown in Table 2.

We also compare the results with a general scheme with mobile sinks that collect data at rendezvous points (RP) [22]. We consider four rendezvous points uniformly distributed in the network in this setting.

Since the sensors do not route the data to the sink or rendezvous points in our approach, the attacker does not gain any extra information about the sink's location by reading the packet header or observing the data traffic. Therefore, the probability for him to locate the sink will be equal to that with a random guess, that is 1/225 in both attack types (1) and (2). On the contrary, data are forwarded to the rendezvous points in the usual RP scheme. The attackers can locate these rendezvous points that the sink will visit by looking at the header field of packets or observing the flow of network traffic. The probability for the attackers to capture the sink in attack types (1) and (2) then becomes 1.0 in RP.

Both attackers in RDC and RP can capture the sink by predicting its next move. Without any information about the sink's previous location, an attacker in RDC has 1/225 probability to predict the sink's next location, since the sink moves randomly among all the nodes in the network. In RP, the sink only moves between the rendezvous points, so the attacker has a probability of 1/4 to predict its next location. If an attacker knows the sink's previous location, e.g., by the information from the captured nodes when the sink passes by them, we find that he has a probability of 0.06 to predict the sink's next location correctly in RDC. We have obtained this probability by running simulations with the sink moving five random steps from its previous location, considering that the attacker will choose the next location that has the highest probability to meet the sink. To include the boundary effects, we also consider the sink starting at random locations and compute the mean probability to predict the next location. With RP, the probability increases to 0.33 when the attacker knows the previous rendezvous point as there are only three remaining rendezvous points left to choose from.

We also measure the time taken for the attacker to meet the sink if he waits at the same location. The waiting time in RDC is much longer than that in RP as the sink may travel to any of the nodes in the network, while, in RP, it walks along a much shorter route visiting only the rendezvous points. Therefore, the attacker in RP can locate and wait for the sink at a rendezvous point in a much shorter and expected time. Overall, the protection strength of RDC is much higher than the traditional RP approach in sensor networks with mobile sinks.

Table 2: Protection Strength Comparison

Attack Types	RDC	RP
Probability to capture the sink by		
(1)Reading packet header	0.004	1.0
(2)Observing network traffic	0.004	1.0
(3a)Predicting the next move without knowing sink's last location	0.004	0.25
(3b)Predicting the next move when knowing sink's last location	0.06	0.33
Waiting time to the next arrival of the sink by		
(4)Staying at the same place	225s	28s

7. CONCLUSIONS

In this paper, we have proposed a random data collection scheme which can protect location privacy of the mobile sinks in WSNs, while providing normal data collection services. Our scheme avoids the location of the sinks to be tracked and protects the sinks from becoming the target of attacks. In our scheme, the sensing data are stored at some random nodes in the network with the sinks moving around randomly and collecting data occasionally from their local neighbors. We have summarized three common kind of attacks threatening location privacy in WSNs and evaluated the protection strength of our proposed scheme. We have analyzed the delivery rate, data collection time and energy consumption of our scheme. We have also evaluated our proposed random data collection scheme by extensive simulations varying different parameters. Both analytical and simulation results show that our scheme can protect the location privacy of mobile sinks effectively, while providing satisfactory data collection services. In the future, we will explore different enhancements to our scheme, like data collection at random time intervals and unequal visiting probability, to further protect the network from smart attackers.

ACKNOWLEDGMENTS

This work was carried out within the Uppsala VINN Excellence Center for Wireless Sensor Networks WISENET, partly supported by VINNOVA, the Swedish Governmental Agency for Innovation Systems.

8. REFERENCES

[1] I. F. Akyildiz, W. Su, and T. Sandarasubramaniam. Wireless sensor networks: a survey. *Computer Networks*, 38(5):393–422, 2002.

[2] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *Elsevier Ad Hoc Networks Journal*, pages 325–349, 2005.

[3] J. Deng, R. Han, and S. Mishra. Countermeasures against trafic analysis attacks in wireless sensor networks. In *Proc. of IEEE/CreateNet International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm)*, 2005.

[4] S. C. Ergen and P. Varaiya. Energy efficient routing with delay guarantee for sensor networks. *ACM Wireless Networks*, 13(5):679–690, Oct 2007.

[5] L. Eschenaur and V. Gligor. A key-management scheme for distributed sensor networks. In *Proc. of the 9th ACM Conference on Computer and Communication Security*, 2002.

[6] M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald. Privacy-aware location sensor netwroks. In *Proc. of USENIX Workshop on Hot Topics in Operation Systems (HotOS IX)*, 2003.

[7] Y. Gu, D. Bozdağ, R. W. Brewer, and E. Ekici. Data harvesting with mobile elements in wireless sensor networks. *Comput. Netw.*, 50(17):3449–3465, 2006.

[8] T. He, J. Stankovic, C. Lu, and T. Abdelzaher. SPEED: a real-time routing protocol for sensor networks. In *Proc. of IEEE ICDCS*, pages 46–55, Providence, RI, U.S., May 2003.

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

[10] D. Jea, A. Somasundara, and M. Srivastava. Multiple controlled mobile elements (data mules) for data collection in sensor networks. In *In DCOSS*, pages 244–257, 2005.

[11] Y. Jian, S. Chen, Z. Zhang, and L. Zhang. Protecting receiver-location privacy in wireless sensor networks. In *Proc. of IEEE Infocom*, pages 1955–1963, 2007.

[12] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk. Enhancing source-location privacy in sensor network routing. In *Proc. of IEEE ICDCS*, Columbus, Ohio, USA, Jun 2005.

[13] B. Karp and H. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proc. of ACM Mobicom*, Boston, Massachusetts, U.S., 2000.

[14] W. Lou and Y. Kwon. H-SPREAD: a hybrid multipath scheme for secure and reliable data collection in wireless sensor networks. *IEEE Trans. on Vechicular Technology*, 55(4):1320–1330, Jul 2006.

[15] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *WSNA '02: Proc. of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97, New York, NY, USA, 2002. ACM.

[16] A. Perrig, R. Szewczyk, D. Tygar, V. Wen, and D. Cullar. Spins: security protocols for sensor networks. *Wireless Communications*, 8(5):521–534, 2002.

[17] R. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: modeling a three-tier architecture for sparse sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 30–41, May 2003.

[18] M. Shao, Y. Yang, S. Zhu, and G. Cao. Towards statistically strong source anonymity for sensor networks. In *Proc. of IEEE Infocom*, 2008.

[19] A. A. Somasundara. Controllably mobile infrastructure for low energy embedded networks. *IEEE Transactions on Mobile Computing*, 5(8):958–973, 2006. Student Member-Kansal,, Aman and Student Member-Jea,, David D. and Fellow-Estrin,, Deborah and Senior Member-Srivastava,, Mani B.

[20] A. A. Somasundara. Mobile element scheduling with dynamic deadlines. *IEEE Transactions on Mobile Computing*, 6(4):395–410, 2007. Member-Ramamoorthy,, Aditya and Senior Member-Srivastava,, Mani B.

[21] W. Wang, V. Srinivasan, and K.-C. Chua. Using mobile relays to prolong the lifetime of wireless sensor networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 270–283, New York, NY, USA, 2005. ACM.

[22] G. Xing, T. Wang, Z. Xie, and W. Jia. Rendezvous planning in mobility-assisted wireless sensor networks. In *RTSS '07: Proceedings of the 28th IEEE International Real-Time Systems Symposium*, pages 311–320, Washington, DC, USA, 2007. IEEE Computer Society.

[23] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao. Towards event source unobservability with minimum network traffic in sensor networks. In *Proc. of ACM WiSec*, Alexandria, Virginia, USA, Apr 2008.

[24] W. Zhang, G. Cao, and T. L. Porta. Data dissemination with ring-based index for wireless sensor networks. *IEEE Transactions on Mobile Computing*, 6(7):832–847, 2007.