Research Article

Protecting Data Source Location Privacy in Wireless Sensor Networks against a Global Eavesdropper

Rong-hua Hu, Xiao-mei Dong, and Da-ling Wang

School of Information Science and Engineering, Northeastern University, Shenyang 110004, China

Correspondence should be addressed to Xiao-mei Dong; dongxiaomei@ise.neu.edu.cn

Received 28 September 2013; Revised 10 February 2014; Accepted 15 July 2014; Published 13 August 2014

Academic Editor: Zujun Hou

Copyright © 2014 Rong-hua Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Data source location privacy (DSLP) is of great importance for some asset monitoring applications in wireless sensor networks (WSNs). Besides the source simulation (SS) method to protect the DSLP against a global eavesdropper in WSNs, other existing methods are based on the panda-hunter game model (PHGM) without considering the communication between data sources and reporter sources, which can cause them to be ineffective. Moreover, there are two limitations in SS. First, the reporter source cannot generate effective event reports. Second, it is unsuitable to track multiobjects accurately. To address the former issue, an improved source simulation (ISS) method is proposed which adjusts the event report strategy. To solve the latter issue, an updated-panda-hunter game model (UPHGM) is proposed and a formal model of the DSLP issues is also presented. Then, based on the UPHGM, an energy-efficient grid-based pull (GBP) scheme is designed to protect the DSLP by combining a light-weight security object collection scheme with an effective grid partition method. Analysis and simulation results show that GBP outperforms SS and ISS in terms of energy cost on the whole.

1. Introduction

A wireless sensor network (WSN) typically consists of a large number of nodes which integrate sensor modules, processing modules, wireless communication modules, and energy modules to sense physical or environmental conditions and collect data. WSNs have promising application prospects in both military and civil domains, such as military surveillance, wildlife habitat monitoring, target tracking, and home automation [1–3]. Compared with traditional networks and ad hoc networks, WSNs have several typical characteristics, such as restricted-resource, no infrastructure, and large number of nodes. These characteristics lead to more challenges in the research of security problems in WSNs [4, 5].

Many works to date in WSN security have focused on providing authentication, confidentiality, integrity, and freshness services. However, for some special applications, it is not enough to provide those security services solely. For example, in the panda-hunter scenario [6], called panda-hunter game model (PHGM), a WSN is deployed to track endangered giant pandas in a vast panda habitat. Each panda carries an electronic tag, called tag-node in this paper, to emit event-trigger-signals, which can be detected by sensors. When a sensor node, called reporter source, detects this signal, it generates an event report and then sends it to a sink node with the help of a security route mechanism. An adversary (the hunter) may locate the monitored pandas, called data sources in this paper, either by detecting the event-trigger-signals or via eavesdropping on the communication between the reporter sources and the sink nodes. Therefore, it is very important to provide the data source location privacy (DSLP) service for this kind of applications.

At present, many efforts have been done to protect the DSLP in WSNs [7, 8]. However, most of them are effective only for local adversaries who merely monitor the local network traffic. For a global adversary, who can monitor the whole network traffic, these approaches would not work. Recently, several schemes have been proposed to protect the DSLP against a global adversary [9–19]. However, most of them except the source simulation (SS) method in [9, 10] ignore the communication between data sources and reporter sources, which makes these approaches ineffective.
Moreover, the SS method also has two limitations. (1) The reporter source cannot generate effective event reports. (2) It is unsuitable to track multiobjets accurately.

In this paper, we focus on protecting the DSLP against a global adversary who can monitor and collect all the messages transmitted in the network at all time. The contributions of this work are summarized in the following.

(i) We point out the existing problems of the source simulation method in [9, 10] and propose an improved source simulation (ISS) method under the panda-hunter game model (PHGM) by adjusting the event report strategy. The ISS method copes with the defect existing in SS, in which the reporter source cannot generate effective event reports.

(ii) We propose an updated-panda-hunter game model (UPHGM), which overcomes the disadvantage of the PHGM and is suitable to monitor multiobjets sources. Also a formal model of the DSLP issues under the UPHGM is given.

(iii) We present an energy-efficient grid-based pull scheme to protect the DSLP under the UPHGM, which includes a light-weight security object collection scheme and an effective grid partition method. It can make sure at wherever a data source lies there is at least one node which can authenticate and collect its related information; meanwhile it can provide security services including confidentiality, authentication, integrity, freshness, and source location privacy. Compared with the source simulation method and the improved source simulation method, it is more energy-efficient.

(iv) We evaluate the energy overhead in both theory and simulation. The results show that (1) the grid-based pull (GBP) method outperforms the SS method and the ISS method on energy overhead on the whole; (2) the energy cost of ISS is comparable to that of SS under low privacy requirement.

The rest of the paper is organized as follows. Section 2 discusses related work. In Section 3, the improved source simulation (ISS) method under the panda-hunter game model (PHGM) is described. In Section 4, the updated-panda-hunter game model (UPHGM) and the formal model of the DSLP issues are presented. In Section 5, the grid-based pull scheme under the UPHGM is proposed. Section 6 describes our evaluations, and Section 7 concludes the paper.

2. Related Work

In [7, 8], extensive surveys have been conducted on the location privacy problem in WSNs. According to adversaries’ capability of monitoring network traffic, existing approaches can be classified into three classes: countering a local adversary [6, 20–28], countering several local adversaries [29], and countering a global adversary [9–19]. In this section, we briefly survey the major related works to provide the DSLP under a global adversary in WSNs.

In [9, 10], two techniques, called periodic collection and source simulation, respectively, were proposed to prevent the leakage of the DSLP. The main idea of the periodic collection method, also called ConstRate scheme in [11], is to make the traffic pattern independent of the presence of real objects. Each node periodically sends packets at a reasonable frequency regardless of whether it has real data to send or not. However, this approach consumes a substantial amount of energy for latency sensitive applications. And it does not consider the traffic generated by electronic tags, which can also be used to infer the locations of monitored objects by the adversary. By simulating the movement patterns of real objects, the source simulation (SS) method creates candidate traces in the network to hide the traffic generated by real objects. It works as follows: before deployment, select L nodes as token node and preload each of them with a unique token ID. After deployment, each token node can mask as a real object to emit event-trigger-signals for event detection. When a node detects the signal, it generates an event report and delivers it to the sink node. The token node then determines the token node of the next round in its neighborhood (including itself) and passes the token to the selected node. In order to prevent the adversary using the token pass messages to distinguish real objects from fake ones, the nodes that detect the real object also need to send a masked token pass message each round. However, this approach has two problems to affect the information collection about real objects. First, as the event-trigger-signals emitted by token nodes and real objects are the same, not only the adversary cannot distinguish from them, but also nodes in the network cannot distinguish from them. As a result, nodes cannot determine whether real event reports or dummy event reports are generated. Second, when there are two or more objects in the network, if a node detects two event-trigger-signals in two rounds, it cannot distinguish whether the signals come from two objects or from one object. Hence, it cannot track multiobjets accurately.

A fitted probabilistic rate scheme, called FitProbRate, was devised in [11] to reduce the event report delay in ProbRate scheme. In ProbRate scheme, the time intervals between message transmissions follow the exponential distribution. In order to reduce the real event report latency, the nodes send the real event reports as soon as possible while keeping the message transmission intervals following the same distribution. Based on the FitProbRate, a distributed resource allocation algorithm was proposed to achieve data source anonymity by mixing real event messages with carefully chosen dummy traffic in [12]. However, the FitProbRate scheme only considers the event indistinguishability, in which the intertransmission times of the real event reports are indistinguishable from the desired distribution of fake transmissions, and ignores the interval indistinguishability, in which the intertransmission times of fake and real intervals are indistinguishable. The adversary can distinguish the real intervals from the fake intervals and then discovers the real objects. In order to solve this problem, an improved method was proposed in [13, 14]. They introduced the same correlation of intertransmission times during real intervals to intertransmission times during fake intervals.
To reduce the traffic caused by the ConstRate scheme in [9, 10] and the ProbRate scheme in [11], based on proxies and data aggregation, a proxy-based filtering scheme (PFS) and a tree-based filtering scheme (TFS) were presented in [15]. In PFS, proxies are selected to collect and filter dummy messages from surrounding nodes. Also proxies are formed into a tree hierarchy and the dummy messages are dropped on the way from proxies to sink nodes. In order to explore the impacts on the network lifetime of proxy filtering approaches, different proxy assignment strategies and different deployment scenarios were investigated in [16]. In [17], a data aggregation scheme based on a cluster-tree hierarchical architecture was proposed to reduce the message overhead by aggregating multiple messages in a single transmission. However, these approaches are also not suited for latency sensitive applications, because when a node detects a real event, it delays the transmission of the real event report such that the next intermessage interval follows the normal application distribution. Similar to periodic collection in [9, 10], these approaches also do not consider the traffic generated by electronic tags.

In [18], four schemes, called naive, global, greedy, and probabilistic, were proposed to protect the DSLP against a global adversary. The naive scheme is similar to the periodic collection method in [9, 10]. Every node periodically sends a real or forge message. The time interval between message transmissions is usually quite long to reduce the communication cost. Therefore, this approach can cause long delay. In order to reduce the delivery latency, the global and the greedy approaches were presented to discover the shortest delivery latency path. However, the global approach requires the knowledge of global network topology and transmission schedules, which are impractical for every node to store the topology of the whole network and to compute the fastest path. The greedy approach is a heuristic method. Every node chooses the neighbor that has the shortest waiting time and is closer to the sink node as the next hop route node. In order to reduce the communication overhead, a probabilistic approach was proposed. This approach lets every node probabilistically transmit messages to guarantee that every node can hear a message during any fixed period of time. However, it ignores that a real object may be detected by some nodes simultaneously, which will generate event reports and lead to more network traffic around the object. This abnormal traffic can help the adversary to discover the real object.

In [19], a spatiotemporal process replication scheme for generating dummy network traffic was proposed to disguise the real event report. The method assumed that monitored events obey a Poisson temporal distribution with a known rate and uniform spatial distribution over the network area. A subset of nodes regularly acts as fake sources and generates dummy network traffic with the same event distribution. However, the method ignores the spatial correlation of events about a real object, which can help the adversary locate the real object. For example, the traces of a panda do not follow uniform spatial distribution over the network during a relatively short period of time. The adversary can locate the panda by observing the traces in a relatively small network area.

3. Improved Source Simulation (ISS)

Method under the PHGM

As mentioned above, all of the prior approaches except the SS method in [9, 10] ignored the event trigger process from tag-nodes. The adversary can defeat all of these approaches by monitoring the additional traffic from tag-nodes. Therefore, to avoid this situation, we should adopt the SS method to simulate the additional traffic pattern generated by tag-nodes. However, as described in Section 2, the SS method has two defects. First, a reporter source cannot generate effective event reports since the event-trigger-signals emitted by token nodes and tag-nodes are indistinguishable. Second, it is not suited to track multiobjects accurately. In this section, we propose an improved source simulation (ISS) method to solve the first problem for single-object monitoring applications. For multiobjects monitoring applications, in order to track the objects effectively and accurately, the event-trigger-signals should contain related information of the monitored objects. The corresponding solving approach will be presented in Section 5.

3.1. Protocol Description. Before deployment, randomly select L nodes as token nodes and preload each of them with a unique token ID. Every token node will mask as a real object to emit event-trigger-signal for event detection. We assume that the infrastructure of secure communications has been established after deployment. Note that, in this approach, the adversary is similar to the one considered in [9–16, 18, 19]; that is, it cannot compromise any sensor node. The event report process includes the following three steps as shown in Figure 1.

(1) Emit Event-Trigger-Signal. A token node or a tag-node emits an event-trigger-signal for event detection around its local area. A node that detects the signal sets itself as a candidate reporter source.

(2) Pass Token. The token node and the candidate reporter sources send the messages called token-pass-msg according to their roles, respectively. We assume that the length of the time interval for passing the token-pass-msg, called token-pass-slot, is $\Delta_{ik}$.

After the token node emits an event-trigger-signal, it sets a timer called timer-for-token-passing with value $T_{ik} \in (0, \Delta_{ik})$. When the timer expires, the token node broadcasts a real token-pass-msg, as follows:

$$\text{token-pass-msg} = \{ \text{ID}_i; t_i; E_{K_{\text{token}}} (\text{DATA1}); E_{K_{\text{rep}}} (\text{DATA2}); \text{MAC1}; \text{MAC2} \},$$

$$\text{DATA1} = \{ \text{token-ID}; t_i \},$$

$$\text{DATA2} = \{ \text{chosen-report-ID}; t_i \},$$
the same length, to prevent the adversary from distinguishing
the real token-pass-msg from the fake token-pass-msg via
packet length. Note that, as the transmission delay is negli-
gible, for example, transmitting one bit information to 50 m
distance only needs 0.17 μs, for the same event, the difference
between \( t_d \) and \( t_s \) is equal to the error of synchronized clocks.
Therefore, we can distinguish different events via comparing
the time field in the token-pass-msgs.

Upon receiving a token-pass-msg, the node ID\(_j\) first
checks the MAC2 using the shared group key \( K_{grp} \) with the
sending node ID\(_i\). If the MAC2 is verified, it sets a flag called
report-flag as true, decrypts the corresponding encrypted
field using the shared group key, and checks the MAC1 using
the pairwise key \( K_{next} \) with node ID\(_i\). If chose-report-ID =
ID\(_j\), it sets a flag called generate-flag as true. If the MAC1 is
verified, it decrypts the corresponding encrypted field using
the pairwise key \( K_{next} \) records the token-ID, and sets itself as
the token node of the next round.

(3) \textbf{Generate and Send Event Reports}. When the token-pass-
slot is over, the token node and the candidate reporter sources
generate and send event reports according to the following
situations.

(i) If the generate-flag of the node ID\(_j\) is true, it means that (1) the event-trigger-signal is emitted by a token
node and (2) it is selected by a token node to generate
the event report. It generates a fake event report and
delivers it to the sink node through an established
routing path using the pairwise key between them.

(ii) If the report-flag of the node ID\(_j\) is false, it means that
the event-trigger-signal is emitted by a tag-node. If
it also does not receive any token-pass-msg for the
same event from a node closer to the sink node, it
generates a real event report and delivers it to the sink
node through an established routing path using the
pairwise key between them.

Compared with source simulation method in [9, 10],
the ISS method has two main advantages. (1) Candidate
report sources can distinguish real events from fake events by
authenticating token-pass-msgs. (2) Only one node generates
an event report for each event-trigger-signal and the amount
of network traffic can be reduced largely.

4. \textbf{UPHGM and Problem Formalization}

Although prior work for protecting the DSLP against a global
adversary was mainly based on the PHGM described in
section one, only the SS method in [9, 10] considered reporter
sources to generate real event reports depending on event-
trigger-signals. It is more realistic than the assumption that
nodes of the protected network can sense the monitored
objects while the adversary cannot. In practice, directly
recognizing an object is a very challenging work due to the
difficulty of distinguishing the physical features of the objects
from background noises [10]. In this section, we will present
an updated-panda-hunter game model (UPHGM). We also
assume that every monitored object is equipped with a sensor
node to emit signals which can be detected by nodes in the network as [9, 10]. The UPHGM includes a network model and an attack model.

4.1. Network Model. We assume that a homogeneous WSN, called obj-WSN, is deployed by an organization to monitor specific objects such as giant pandas, as shown in Figure 2. The obj-WSN consists of \( N \) common nodes and one sink node. All of the common nodes have roughly the same resources. Each common node \( u \) has a unique node identifier (ID) and knows its own location. The communication radius of every common node is \( R \). Since directly recognizing the object is very difficult, in order to avoid directly sensing the objects, every monitored object is embedded a sensor node, that is, tag-node, to emit event-trigger-signals, called Object-msg. A reporter source which receives an Object-msg will generate a real event report and transmit it to the sink node according to a chosen route mechanism. The communication radius of every tag-node is \( r \). To distinguish different objects, each tag-node \( v \) also has a unique tag identifier (TID). Each Object-msg contains the TID of the tag-node and other information of interest (interest-info).

4.2. Attack Model. We assume that the adversary has his/her own sensor network deployed in the same area, as shown in Figure 2, to monitor the global network traffic of the obj-WSN as [9, 10]. Note that the deployment time of the adversary’s network is later than that of the obj-WSN. More specifically, we assume that when the infrastructure of secure communications of the obj-WSN has been established, then the adversary’s network can be deployed. The adversary can monitor the whole network traffic including the Object-msgs emitted by tag-nodes. Knowing a global view of the network traffic, the adversary can easily deduce where the objects are moving around. For example, an object is very likely close to tag-nodes and reporter sources. We do not consider the situations that the monitored objects can be directly recognized by sensors. If that happens, then the adversary can launch the direct sensing attack and any defense mechanism cannot protect the location privacy of the monitored objects.

In addition, the adversary has the following characteristics.

(1) To appropriately study privacy, we apply Kerckhoff’s principle [30]. We assume that the adversary knows the communication protocols and defense mechanisms of the obj-WSN. Each eavesdropping node knows its own location, as shown in Figure 2.

(2) To be invisible from obj-WSN, the adversary considered in this paper may launch only passive attacks and avoid active attacks as [9, 10]. However, since the network may also be attacked by other adversaries with the different attack aims, we also need to prevent them from launching some active attacks such as injecting bogus data by utilizing the security weaknesses of the defense mechanisms.

4.3. Distinguish with the PHGM. The main difference between UPHGM and PHGM is that the former considers the Object-msg containing the TID of the tag-node and other interest-info, which is more suitable for many practical applications. For example, for multiobjects monitoring applications, if Object-msg contains no information of a monitored object and is only used to trigger event detection, a reporter source cannot generate an accurate event report for the corresponding object, because it does not know which tag-node emits the received Object-msg. And for many applications, we may want to know the state information or health characteristic information about monitored objects. Therefore, the UPHGM is more realistic and effective than the PHGM for many applications to monitor multiobjects.

Compared with the PHGM, the UPHGM yields a new problem which needs to be solved. That is, how to forward Object-msgs to the sink node securely while providing location privacy to tag-nodes and reporter sources under a global adversary. Prior work only considered the location privacy of reporter sources and ignored the location privacy of tag-nodes. Although reporter sources can use the established pairwise keys to communicate with common nodes securely, due to tag-nodes being of moving characteristic and their resource constraints, the tag-nodes cannot firstly establish pairwise keys with all common nodes in the network and then use the corresponding keys to communicate with the corresponding nodes securely. Therefore, secure and effective protocols have to be designed to implement the secure communications between tag-nodes and reporter sources while providing their location privacy.

4.4. Formal Model. The problem \( \Omega \) for protecting the source location privacy against a global adversary in WSNs can be represented by an eight-tuple \((O, Net_G, Net_A, D, C_D, A, S_A, \text{and } b)\), where one has the following.
(1) $O$ is the set of monitored objects. We assume that the size of $O$ is $M$. Every monitored object can move around in the deployment area.

(2) $Net_{q}$ is the monitoring network deployed by a valid organization such as the obj-WSN.

(3) $Net_{a}$ is the hunting network deployed by an adversary to eavesdrop on the network transmissions of $Net_{q}$.

(4) $D$ is the set of defense strategies to protect the source location privacy.

(5) $C_{D}$ is the cost of defense strategies, mainly consisting of energy cost.

(6) $A$ is the set of attack strategies to infer the locations of monitored objects.

(7) $S_{A}$ is the set of candidate objects in whose range the adversary expects to find the monitored objects.

(8) $b$ is the level of privacy in terms of bits. It is defined as

$$b = -\sum_{|S_{A}|} \frac{1}{\log S_{A}} = \log \frac{|S_{A}|}{M},$$

(3)

where $|S_{A}|$ and $|O|$ are the size of $S_{A}$ and $O$, respectively.

Given the level of privacy $b$, our goal is to achieve it with minimum defense overhead $C_{D}$.

5. Grid-Based Pull (GBP) Scheme under the UPHGM

As we have pointed out in Section 4, the UPHGM is more realistic than the PHGM and is suitable to monitor multi-objects accurately. However, there are still more challenges. In this section, we will propose a grid-based pull scheme to protect the DSLP under the UPHGM by combining a lightweight security object collection scheme with an effective grid partition method. It includes selecting communication scheme and determining grid size. And its implementation is then presented.

5.1. Selecting Communication Scheme. The first main challenge is to effectively authenticate and communicate between a mobile tag-node and each potential communication node in the network. There are two kinds of communication schemes: push scheme and pull scheme. For push scheme, each tag-node broadcasts Object-msg periodically. As the tag-nodes are mobile, every node in the network has to be able to authenticate each tag-node. For pull scheme, the query nodes broadcast query messages periodically. In this case, each tag-node must be able to authenticate every node in the network.

Compared with the push scheme, the pull scheme is more secure to solve our problem. For the push scheme, if any node is compromised by the adversary, he or she can obtain the secret information such as secure keys stored in a compromised node and then upload the obtained secret information to every eavesdropping node. As a result, when any tag-node broadcasts an Object-msg, eavesdropping nodes can authenticate the message and recognize the monitored object.

For pull scheme, if any node is compromised by the adversary, to launch the stealthy attacks, he or she shall not broadcast query message using the obtained secure information. If he or she does so, then it is very probably to expose the eavesdropping nodes to the obj-WSN and to trigger alarms. Only in this situation, when a monitored object lies in the communication range of the compromised node and the compromised node is just the query node, the location of the monitored object is exposed to the adversary without triggering alarms. Therefore, as shown in Table 1, considering security, we should select the pull scheme.

However, due to resource constrains, for example, the storage resource, in order to let a tag-node be able to authenticate every node in the network, obviously, we cannot select symmetric approaches, which require the tag-node store a large number of pairwise keys. Hence, for storage efficiency, we should adopt an asymmetric authentication approach. Considering the asymmetric feature between tag-nodes and common nodes in the obj-WSN, we design an authentication approach based on the identity-based cryptography (IBC), which will be described in Section 5.3.

5.2. Determining Grid Size. The second problem we consider is how to reduce the number of query nodes as largely as possible while keeping the high quality tracing of the monitored objects. To reduce the number of query nodes, we divide the whole network area into a number of virtual grids. And in each grid a node, called duty node, is selected to collect information of the monitored objects. To balance energy load among nodes, each node can be responsible for collecting the information for a certain time interval in turn as duty node. Hence, the second problem can be translated into the problem that how to determine the size of the virtual grid to satisfy both query requirement and connectivity requirement. The query requirement means that no matter where a tag-node is in the network, its relevant information can be detected by at least one query node. The connectivity requirement means that nodes in a grid are capable of communicating with those nodes in an adjacent grid.

On the one hand, to meet the query requirement, there should be at least one full grid in the circle with the tag-node as the center and with the communication radius $r$ of the tag-node as the radius, just as illustrated in Figure 3. Obviously,
the smaller the size of the grid is, the more full grids are in the circle. To reduce the number of query nodes as largely as possible, we need to determine the maximum size of the grid. We have derived the theoretical maximum size of the grid and have the following theorem.

**Theorem 1.** To ensure that no matter where a tag-node is in the network, there is at least one full grid in the circle $C_r$ with the tag-node as the center and with the communication radius $r$ of the tag-node as the radius, the maximum length $a$ of the grid is $r/\sqrt{2}$.

**Proof.** Firstly, we prove that there is at least one full grid in the circle $C_r$ when $a = r/\sqrt{2}$. We assume that the circle $C_r$ does not contain any full grid. Obviously, in this case, the center of the circle is at the common vertex of four grids most likely, as shown in Figure 4. In this case, the circle $C_r$ just contains four full grids, as shown in Figure 4. That is, the assumption does not hold. In other words, there is at least one full grid in the circle $C_r$.

Secondly, we prove that the $a$ is the maximum length. We assume that the length $a^* = a + \epsilon$ ($\epsilon > 0$) also satisfies the requirement in the theorem. However, when the center of the circle is at the common vertex of four grids, the length of the diagonal line of one grid $d_4 = \sqrt{2}a^* = \sqrt{2}(a + \epsilon) = \sqrt{2}(r/\sqrt{2} + \epsilon) > r$, the circle $C_r$ does not contain any full grid, which is paradox with the assumption, as shown in Figure 4. That is, the assumption does not hold. This proves that the $a$ is the maximum length.

On the other hand, according to [31], to satisfy the connectivity requirement, the length $a$ of the grid should satisfy $a \leq R/\sqrt{5}$, where $R$ is the communication radius of every common node, as shown in Figure 5.

As the communication radius of the common node is usually not less than that of the tag-node, we assume that

$$R = \lambda r,$$

where $\lambda \geq 1$. Let $l$ be the length of the network. Define $m$ and $n$ by

$$m = \frac{l}{(r/\sqrt{2})}, \quad n = \frac{l}{(R/\sqrt{5})}. \quad (4)$$

That is, $m$ and $n$ are the number of grids according to the above two dividing methods, respectively. Then, we have the following conclusions.

(i) If $m = n$, that is, $\lambda = \sqrt{2.5}$, the above two dividing methods are the same.

(ii) If $m > n$, that is, $\lambda > \sqrt{2.5}$, to satisfy the above two requirements, we should select the former dividing method.

(iii) If $m < n$, that is, $1 \leq \lambda < \sqrt{2.5}$, to satisfy the above two requirements, we should select the latter dividing method.

5.3. Implementation of the Grid-Based Pull Scheme. In this subsection, we introduce the implementation of the grid-based pull scheme, including predeployment phase, grid formation phase, and report collection phase.

5.3.1. Predeployment Phase. Before deploying the obj-WSN, we assume that a trusted authority (TA) does the following operations.
(1) Generate system parameters \((p, q, E/F_p, G_a)\), where \(p, q\) are two large primes, \(E/F_p\) indicates an elliptic curve \(y^2 = x^3 + ax + b\) over the finite field \(F_p\), and \(G_a\) is a \(q\)-order subgroup of the additive group of points of \(E/F_p\).

(2) Choose two hash functions: \(H : \{0, 1\}^* \rightarrow G_a\), mapping strings to nonzero elements in \(G_a\), and \(h\), mapping arbitrary inputs to fixed-length outputs, for example, SHA-1 [32].

(3) Select a random element \(k \in Z_q^* = \{x \mid 1 \leq x \leq q - 1\}\) as the network master secret.

(4) Calculate the \(H(\text{ID}_u) = (X_u, Y_u)\) and the IBC key \(IK_u = kH(\text{ID}_u) \in G_a\) for each common node \(u\). Note that, due to the elliptic curve discrete logarithm problem (ECDLP), it is computationally infeasible to obtain \(k\) from any \((IK_u, \text{ID}_u)\) pair.

(5) Randomly select \(L\) common nodes as token nodes and preload each of them with a unique token ID.

Each common node \(u\) is preloaded with the public system parameters \((p, q, E/F_p, G_a, H, \text{ and } h)\), its private key \(IK_u\), its public key \((X_u, Y_u)\), and a pairwise key with the sink node \(K_{uv}\). Each tag-node \(v\) is preloaded with the same public system parameters as each common node, the network master secret \(k\), and a pairwise key with the sink node \(K_{v:br}\).

5.3.2. Grid Formation Phase. After deploying the obj-WSN, we assume that every common node can obtain its own location information using some localization methods such as [33, 34]. Also we assume that every common node can obtain the location information \((x_{br}, y_{br})\) of the sink node either via preloading method or via broadcasting method. The grid identifier \(G_u = (g_x,u, g_y,u)\) of node \(u\) can be calculated using the following:

\[
g_x,u = \left(\frac{x_u - x_{br}}{a}\right), \quad (5)
\]

\[
g_y,u = \left(\frac{y_u - y_{br}}{a}\right),
\]

where \(x_u\) and \(y_u\) indicate the geographic location of node \(u\) and \(a\) is the side length of a grid, which is determined according to Section 5.2.

To form virtual grids, in network initialization phase, each common node \(i\) broadcasts a hello message (hello-msg), as follows:

\[
\text{hello-msg} = \{\text{ID}_i; x_i; y_i\}.
\]

Upon receiving the hello-msg from node \(i\), the node \(u\) first calculates the grid identifier \(G_i\) of node \(i\) using the \((x_i, y_i)\) information. If \(G_i = G_u\), it then puts the information \((\text{ID}_i; x_i; y_i)\) to its same-grid table in descending order of ID number; otherwise, it puts the information \((\text{ID}_i; x_i; y_i)\) to its non-same-grid table in increasing order of distance to the sink node.

To avoid additional communication overhead caused by the duty node role rotation, we can stipulate the rotation order, for example, in descending (or increasing) order of ID number, in each grid. Thus, each grid will have a duty node when the network initialization finishes and also each node will rotate as duty node in a stipulated order.

5.3.3. Report Collection Phase. We assume that in network initialization phase an effective key-based mechanism is adopted, such as [35–37]. That is, an infrastructure for secure communication has been established. To achieve anonymity, nodes may need to use pseudonyms to communicate with each other according to the requirement. We adopt the method in [29] to generate pseudonyms. Assuming two nodes \(u\) and \(v\) with a shared key \(K_{uv}\), we can generate a sequence of pseudonyms using the hash function \(h\) by iteratively hashing a random value \(V_{rmd}\), as follows:

\[
id_{uv}^{(1)} = h(K_{uv} || V_{rmd}),
\]

\[
id_{uv}^{(i)} = h(K_{uv} || \text{id}_{uv}^{(i-1)}).
\]

(1) Broadcast Query Message. After network initialization, every duty node \(u\) periodically broadcasts query message (query-msg), as follows:

\[
\text{query-msg} = \{\text{ID}_u; t_{qj}; X_u; Y_u; MAC_{IK_u}(ID_v; t_{qj}; X_u); MAC_{K_{qj}}(ID_u; t_{qj}; X_u)\},
\]

where \(t_{qj}\) is the timestamp and its value is the time when the query-msg is broadcast by the duty node; \(K_{qj}\) is the group key shared among the sending node and its neighborhood; and \(X_u\) is the \(X\) coordinate of the node \(u\) on the elliptic curve \(E/F_p\). Since \(H(\text{ID}_u) = (X_u, Y_u)\) is a point of \(E/F_p\), only one of \(X_u\) and \(Y_u\) coordinates needs to be transmitted and the other can be easily derived using the curve equation.

(2) Reply Query Message

(i) For Tag-Node. We assume that tag-nodes and common nodes are loosely synchronized. Every tag-node periodically monitors the query-msg for a specific time interval \(\Delta_m\), called monitoring slot. Each monitoring period is called a monitoring round. Upon receiving a query-msg from a duty node \(u\), a tag-node \(v\) performs the following operations.

(I) Calculate the \(Y_u\) using \(X_u\) and the curve, and then calculate the IBC key \(IK_u = k(X_u, Y_u) \in G_a\) using the \((X_u, Y_u)\) and the network master secret \(k\).

(II) Check the first MAC in (9): calculate \(MAC' = h_{IK_u}(\text{ID}_u || t_{qj} || X_u)\) and compare whether the received corresponding MAC is the same as the \(MAC'\). If the MAC is verified, it puts the \(\text{ID}_u\) into a candidate reply-list and stops receiving the query-msg in current monitoring slot; otherwise, it ignores the query-msg.
(iii) Send a reply message (reply-msg). The tag-node \( v \) selects a node in its candidate reply-list to reply. We assume that query node \( u \) is the candidate node; then tag-node \( v \) sends a reply-msg to node \( u \), as follows:

\[
\text{reply-msg} = \left\{ \text{ID}_{v_u}; \text{id}^{(*)}_{v_u}; \text{id}^{(*)}_{v \text{-next}}; E_{K_{v \text{-next}}} (\text{token-ID}; t; \text{others1}) ; \right.
\]
\[
\left. \quad \text{MAC}_{K_{v \text{-next}}} (\text{DATA1}) ; \text{MAC}_{K_u} (\text{DATA2}) \right\},
\]
where \( \text{id}^{(*)}_{v_u} \) is the pseudonym shared between tag-node \( v \) and the sink node, \( \text{ID}_{v_u} \) is the tag ID of tag-node \( v \), and \( t \) is the sending timestamp. After that, \( v \) updates the corresponding pseudonym using (8).

(ii) For Token Node. In the same manner as tag-nodes, when a token node \( w \) receives a query-msg from a duty node \( u \), which, we assume, is the candidate node to reply, it sends a reply-msg to node \( u \). Note that token nodes do not perform the operations (I) and (II) as tag-node does do. A token node just checks the second MAC in (9) with the corresponding shared group key. Token node \( w \) sends the following reply-msg to node \( u \):

\[
\text{reply-msg} = \left\{ \text{ID}_{u}; \text{id}^{(*)}_{u \text{-next}}; \text{id}^{(*)}_{w_u}; E_{K_{w \text{-next}}} (\text{token-ID}; t; \text{others2}) ; \right.
\]
\[
\left. \quad \text{MAC}_{K_{w \text{-next}}} (\text{DATA1}) ; \text{MAC}_{K_{u \text{-next}}} (\text{DATA2}) \right\},
\]
\[
\text{DATA1} = \left\{ \text{id}^{(*)}_{u \text{-next}}; E_{K_{u \text{-next}}} (\text{token-ID}; t; \text{others2}) \right\},
\]
\[
\text{DATA2} = \left\{ \text{id}^{(*)}_{w_u}; E_{K_{w_u \text{-next}}} (\text{token-ID}; t; \text{others2}) \right\},
\]
where \( \text{id}^{(*)}_{u \text{-next}} \) and \( K_{w \text{-next}} \) are the pseudonym and key shared between token node \( w \) and the next round’s token node (nextTkNode), respectively, which is selected by token node \( w \) in its neighborhood (including itself), \( \text{id}^{(*)}_{w_u} \) and \( K_{w_u} \) are the pseudonym and key shared between token node \( w \) and query node \( u \), respectively, and \( t \) is the sending timestamp. After that, \( w \) updates the corresponding pseudonyms using (8).

As the reply-msg sent from token node \( w \) contains the pseudonym shared between \( w \) and nextTkNode, hence, nextTkNode also receives the reply-msg. When receiving this message, it first checks the corresponding MAC using the pairwise key shared between \( w \) and itself. If the corresponding MAC is verified, it updates the pseudonym shared between \( w \) and itself using (8), decrypts the corresponding encrypted field, obtains the token-ID, and sets itself as the next round’s token node.

(3) Generate and Forward Event Report Message. Due to the different schemes used by tag-nodes and token nodes to generate pseudonyms, as shown in (II) and (8), a query node which receives a reply-msg can know whether the message is sent from a tag-node or a token node. Therefore, the query node can check the MAC using the corresponding key. Upon receiving a reply-msg from a tag-node \( v \) or a token node \( w \), a query node \( u \) first checks the corresponding MAC. If the corresponding MAC is verified, it generates an event report (rpt-msg) and forwards the rpt-msg to the duty node closest to the sink node in its communication range.

Query node \( u \) forwards the following real rpt-msg to the chosen next hop node (nextHop) if it receives a verified reply-msg from tag-node \( v \):

\[
\text{rpt-msg} = \left\{ \text{ID}_{u}; \text{ID}_{\text{nextHop}}; t_{rs}; \text{ID}_{rs}; \right.
\]
\[
\left. \quad E_{K_{rs \text{-bs}}} (\text{collectInfo}) ; \text{MAC}_{K_{u \text{-nexthop}}} (\text{DATA}) \right\},
\]
\[
\text{DATA} = \left\{ \text{id}^{(*)}_{u \text{-next}}; E_{K_{u \text{-next}}} (\text{collectInfo}) \right\},
\]
where \( \text{id}^{(*)}_{u \text{-next}} \) is the ID of the reporter source, that is, the first reporter node and in this case its tag-node \( u \), \( t_{rs} \) is the sending timestamp from the reporter source, and collectInfo is shown in (14).

If query node \( u \) receives a verified reply-msg from token node \( w \), it forwards the following fake rpt-msg to the chosen next hop node (nextHop):

\[
\text{rpt-msg} = \left\{ \text{ID}_{u}; \text{ID}_{\text{nextHop}}; t_{rs}; \text{ID}_{rs}; \right.
\]
\[
\left. \quad \text{rdmDATA} ; \text{MAC}_{K_{u \text{-nexthop}}} (\text{DATA}) \right\},
\]
\[
\text{DATA} = \left\{ \text{id}^{(*)}_{u \text{-next}}; E_{K_{u \text{-next}}} (\text{collectInfo}) \right\},
\]
where \( \text{rdmDATA} \) is a random binary sequence code and its length shall satisfy the fact that all rpt-msgs have the same length, to prevent the adversary from distinguishing the real rpt-msgs from the fake ones via packet length.

When a relay-node receives a rpt-msg, it checks the MAC. If the MAC is verified, the relay-node regenerates MAC using the pairwise key shared with the next hop node. Then it forwards the rpt-msg to the next hop node as quickly as possible whenever the channel is free. In this way, the rpt-msg will eventually reach the sink node.

When the sink node receives a verified rpt-msg generated from a reporter source \( u \), it first obtains the collectInfo field via decrypting the encrypted part using the key shared between \( u \) and itself with the help of the ID_{rs} field in (18). Then it can verify the collectInfo and decrypt the encrypted field in (12) using the key shared with the tag-node if the pseudonym field in (12) of the received rpt-msg is the pseudonym of a tag-node. Therefore, the sink node can easily obtain the interesting information about the monitored objects.
5.3.4. Security Analysis. The GBP scheme provides the following security services.

(1) Confidentiality Service. The confidentiality service provided by the GBP scheme is shown in two aspects. On the one hand, the GBP scheme achieves the confidentiality of identifications of the tag-nodes and the token nodes at the reply query message phase via the pseudonym technique, as shown in (10) and (15). In other words, when receiving a reply-msg, the adversary cannot know whether the message is sent by a tag-node or a token node from the pseudonym fields, that is, from id\(_{v_{bs}}^\ast\) and id\(_{w_{next}}^\ast\). This is because the adversary does not have the corresponding secret keys K\(_{v_{bs}}\) and K\(_{w_{next}}\) to generate corresponding pseudonyms, as shown in (22) and (23). However, query nodes and the chosen next round’s token nodes can receive reply-msgs effectively with the help of the corresponding secret keys such as IK, K\(_{w}\), and K\(_{w_{next}}\), as shown in (II), (24), and (23). If the corresponding pseudonym is id\(_{w}\), it implies that the reply-msg is sent from a tag-node; if the corresponding pseudonym is id\(_{w_{next}}\), it implies that the reply-msg is sent from token node w. And if a node has the pseudonym id\(_{w_{next}}\), it implies that it is the chosen token node of next round. It not only guarantees that tag-nodes reply the query messages confidentially, but also guarantees that token nodes pass tokens confidentially:

\[
\begin{align*}
    id_{v_{bs}}^\ast &= h(K_{v_{bs}}\|id_{v_{bs}}^{\ast-1}), \\
    id_{w_{next}}^\ast &= h(K_{w_{next}}\|id_{w_{next}}^{\ast-1}), \\
    id_{w}^\ast &= h(K_{w}\|id_{w}^{\ast-1}).
\end{align*}
\]

However, using the pseudonym technique we have to solve the influence of pseudonym collision; that is, more than one node has the same pseudonym, because the hash function may generate same outputs from different inputs. The birthday paradox implies that the pseudonym collision probability is \(2^{-n_{bit}/2}\), where \(n_{bit}\) is the number of bits of a pseudonym. Given an 8-byte pseudonym, then, the collision probability is \(2.3e^{-10}\), which implies that this probability can be negligible. At the same time, as we also provide the authentication service via checking MACs, even though a pseudonym collision happens, a node can judge whether a received message is delivered to it or not. In other words, using 8-byte pseudonyms and with the help of the authentication service, we can guarantee that tag-nodes reply query messages and token nodes pass tokens unambiguously.

On the other hand, the GBP scheme guarantees confidentiality of the contents of messages via encrypting their contents using secret pairwise keys K\(_{v_{bs}}\), K\(_{w_{next}}\), and K\(_{rs_{bs}}\), as shown in (10), (15), and (18).

(2) Authentication Service. The authentication service of the GBP scheme is provided via checking MACs and pseudonyms. For example, in (9), tag-nodes and token nodes can authenticate duty nodes via using corresponding keys such as IK, and K\(_{w}\) to check the first MAC MAC\(_{1}\) and the second MAC MAC\(_{2}\), respectively, as shown in (25) and (26).

Obviously, if the corresponding MAC is verified, it implies that the corresponding duty node owns the required key, and the authentication is verified. In order to defend against an adversary forging a valid MAC for a particular message, we adopt a 4-byte MAC as the TinySec-AE [38] packet does. This implies that an adversary has a 1 in \(2^{32}\) chance in blindly forging a valid MAC for a particular message, which can provide an adequate level of security for WSNs [38]. Similarly, duty nodes and chosen next round’s token nodes can authenticate tag-nodes and current token nodes via checking MACs; relay-nodes can authenticate each other using MACs. And duty nodes and chosen next round’s token nodes can authenticate tag-nodes and current token nodes via checking pseudonyms, as shown in (11), (24), and (23):

\[
\begin{align*}
    MAC^{(1)} &= h_{IK_u} (ID_u||t_Q||X_u), \\
    MAC^{(2)} &= h_{K_{w}} (ID_u||t_Q||X_u).
\end{align*}
\]

(3) Integrity Service. The integrity service of the GBP scheme is provided via checking MACs similar to the analysis for the authentication service, such as in (9), (10), and (15).

(4) Freshness Service. The freshness service of the GBP scheme can be provided via the timestamp in (9), (18), and (20). For example, if a tag-node receives a query-msg, it can compare the timestamp field \(t_Q\) in (9) with its local time \(t_{self}\) as (27). If \(\Delta_t\) is above a certain threshold \(\Delta_{t\_fresh}\), we can believe that the received query-msg is a replay message and the tag-node shall discard the message. If \(\Delta_t \leq \Delta_{t\_fresh}\), then the tag-node will check the first MAC field in (9), as shown in (25). If the MAC field is incorrect, the received query-msg may be a forged message using a fresh timestamp. Else we can believe the received message is fresh. At the same time, the freshness service of the GBP scheme can be provided by the second pseudonym field in (10) and (15). In (10), the pseudonym id\(_u\) is generated with the help of the query timestamp \(t_Q\), as shown in (11). When a query node u receives a reply-msg from a tag-node, it can judge whether the received message is fresh or not via comparing the pseudonym field id\(_u\). In (15), if a query node u receives a reply-msg from a token node w, it can also judge whether the received message is fresh or not via comparing the pseudonym field id\(_w\). If they own the synchronous pseudonym, then it can believe the received message is fresh:

\[
\Delta_t = |t_Q - t_{self}|.
\]

(5) Privacy Service. As both the adversary and the defender have similar knowledge about the behavior of real objects, we assume that any candidate trace created by the token nodes in the network will be considered as a valid candidate trace by the adversary as [9, 10]. On the one hand, as the confidentiality service is provided by the GBP scheme, hence, the adversary cannot distinguish between tag-nodes and token nodes based on the contents of messages in the network. On the other hand, as the communication
schemes of tag-nodes and token nodes are the same and indistinguishable for the adversary, hence, the adversary cannot distinguish between tag-nodes and token nodes based on the communication schemes. That is, the adversary cannot use the content information and the contextual information to distinguish tag-nodes from token nodes. In other words, L token nodes create L valid candidate traces. Therefore, we can see that the set of candidate locations includes an average of \( (M + L) \) candidate objects; that is, \( |S_M| = (M + L) \). As a result, according to (3), the privacy provided by the GBP scheme can be estimated by

\[
b = \log_2 \frac{M + L}{M}.
\]

Let us have a look at the following example where we have one monitored object, \( M = 1 \), and one fake object, \( L = 1 \), in the network. Then, according to (28), \( b = 1 \). It means that one of the two candidate objects is the monitored object, since one bit can denote two possibilities.

6. Evaluations

In this section, we first give the communication and computation overhead of both our schemes and the SS scheme in [9, 10]. Then, we present performance results of them in terms of communication overhead.

6.1. Numerical Discussion. In this subsection, we provide mathematical analysis to get the numerical results of both our methods and the SS scheme in [9, 10], in terms of communication overhead and computation overhead.

6.1.1. Packet Format and Length. In order to analyze the communication cost accurately, in this section, we provide the packet format and length for each type of messages. According to IEEE 802.15.4 frame format [39], each packet includes 9 bytes constant information, which consists of the 4-byte preamble sequence, the 1-byte start of frame delimiter, the 1-byte frame length, the 2-byte frame control field, and the 1-byte data sequence number. That is to say, each type of messages in both our schemes and SS includes the 9 bytes constant information.

In GBP, a query-msg carries a 2-byte source address, which is denoted by the identification of the broadcasting node, a 4-byte timestamp, a 64-byte X coordinate of the broadcasting node on the elliptic curve \( E/F_p \), and two 4-byte MACs in addition to the 9-byte constant information. A 2-byte source address means that the network can contain \( 2^{16} - 1 = 65535 \) nodes and may be sufficient for most applications. A 64-byte X coordinate means that \( p \) is a 512-bit prime. Let the order \( q \) of \( G_p \) be a 160-bit prime. According to [40, 41], the chosen parameters deliver an equivalent level of security to that of 1024-bit RSA. We denote the byte length of the query-msg by \( \text{len}_{\text{query-msg}} \). So, we have \( \text{len}_{\text{query-msg}} = 87 \).

In GBP, a reply-msg consists of the 9-byte constant information, two 8-byte pseudonym fields, an encrypted field, and two 4-byte MACs fields. We assume that the encrypted function is implemented using the AES-128 with a 16-byte output. Then, the byte length of the reply-msg \( \text{len}_{\text{reply-msg}} \) = 49.

In GBP, a rpt-msg carries three 2-byte ID fields, a 4-byte timestamp, an encrypted field, and a 4-byte MAC field in addition to the 9-byte constant information. As shown in (12) and (14), since the byte length of the encrypted information \( \text{collectInfo} \) is 28, including an 8-byte pseudonym, a 16-byte encrypted field, and a 4-byte MAC, the byte length of the encrypted field of the rpt-msg is 32 using the AES-128 with two 16-byte outputs. Therefore, the byte length of the rpt-msg \( \text{len}_{\text{rpt-msg-GBP}} \) = 55 in GBP.

In both ISS and SS, we assume that an event-trigger-signal carries a 4-byte timestamp and a 4-byte MAC in addition to the 9-byte constant information. Thus, the byte length of the event-trigger-signal \( \text{len}_{\text{trig-msg}} \) = 17.

In ISS, a token-pass-msg consists of the 9-byte constant information, a 2-byte ID field, a 4-byte timestamp, two encrypted fields, and two 4-byte MACs fields. Similar to the encrypted field of the reply-msg in GBP, the byte length of the encrypted field of the token-pass-msg in ISS is 16. Therefore, the byte length of the token-pass-msg \( \text{len}_{\text{token-pass-msg-iss}} \) = 55 in ISS.

In SS, a token-pass-msg carries a 2-byte ID field, a 4-byte timestamp, a 16-byte encrypted field, and a 4-byte MACs field in addition to the 9-byte constant information. Thus, the byte length of the token-pass-msg \( \text{len}_{\text{token-pass-msg-ss}} \) = 35 in SS.

In both ISS and SS, similar to in GBP, a rpt-msg consists of the 9-byte constant information, three 2-byte ID fields, a 4-byte timestamp, a 16-byte encrypted field, and a 4-byte MAC field. Therefore, the byte length of the rpt-msg \( \text{len}_{\text{rpt-msg-iss}} \) = 39 in ISS and SS.

6.1.2. Communication Overhead. In GBP, the communication overhead includes broadcasting and receiving query-msg overhead, replying and receiving query-msg overhead, and sending and receiving event report message (rpt-msg) overhead. (1) In the broadcasting query-msg phase, duty-nodes broadcast query-msgs, and tag-nodes and token nodes listen on the chosen channel. Note that in order to reduce the interference caused by broadcasting query-msg and replying query-msg, we can transmit them on two different channels. Therefore, the number of query-msg broadcast, \( N_{\text{Tx,qry-msg}} \), in the network will be equal to the number of the duty nodes; the number of query-msg received, \( N_{\text{Rx,qry-msg}} \), in the network will be equal to the number of token nodes (fake objects) plus tag-nodes (real objects). Suppose that the sensor field is divided into \( n_{\text{row}} \times n_{\text{clm}} \) square grids. That is, the number of duty nodes is \( n_{\text{row}} \times n_{\text{clm}} \). Thus, \( N_{\text{Tx,qry-msg}} = n_{\text{row}} \times n_{\text{clm}} \). Assume that the number of fake objects is \( L \) and the number of real objects is \( M \); then \( N_{\text{Rx,qry-msg}} = (L + M) \).

(2) In the replying query-msg phase, tag-nodes and token nodes send reply-msgs, and their neighbor nodes receive reply-msgs. Therefore, the number of reply-msgs transmitted, \( N_{\text{Tx,rpl-msg}} \), in the network will be equal to the number of token nodes and tag-nodes; that is, \( N_{\text{Tx,rpl-msg}} = (L + M) \); the number of query-msg received, \( N_{\text{Rx,rpl-msg}} \), in the network will be equal to the number of their neighboring nodes. Let \( \rho_{\text{node}} \) denote the average number of neighboring
nodes of one node in the network. Then, \( N_{\text{RX-rpl-msg}} = (L + M) \rho_{\text{node}} \).

(3) In the sending rpt-msgs phase, the chosen duty-nodes generate rpt-msgs and deliver them to the sink node with the help of intermediate duty nodes along the forwarding paths. Let \( \text{hop}_{\text{avg}} \) be the average hop distance from a reporter source to the sink node, and let \( N_{\text{TX-rpt-msg-gbp}} \) and \( N_{\text{RT-rpt-msg-gbp}} \) be the number of rpt-msgs transmitted and received in the network, respectively. Since there are \( L + M \) reporter sources, and every reporter source generates a rpt-msg and delivers it to the sink node, we have \( N_{\text{TX-rpt-msg-gbp}} = N_{\text{RX-rpt-msg-gbp}} \) and \( N_{\text{TX-rpt-msg-gbp}} = (L + M) \text{hop}_{\text{avg}} \). Table 2 shows the total number of messages transmitted and received in each phase in the GBP scheme. Let \( E_{\text{TX}} \) and \( E_{\text{RX}} \) be the energy consumption for transmitting and receiving one byte, respectively. And let \( E_{\text{TX-GBP}} \) and \( E_{\text{RX-GBP}} \) be the energy consumption for transmitting and receiving messages in the GBP scheme, respectively. Then, we have

\[
E_{\text{TX-GBP}} = (N_{\text{TX-qry-msg}} \text{len}_{\text{query-msg}} + N_{\text{TX-rpl-msg}} \text{len}_{\text{reply-msg}} + N_{\text{TX-rpt-msg-gbp}} \text{len}_{\text{rpt-msg-gbp}}) E_{\text{TX}} = E_{\text{TX}} \times n_{\text{row}} \times n_{\text{clm}} \times 87 + E_{\text{TX}} \times (L + M) \times 49 + E_{\text{TX}} \times (L + M) \times \text{hop}_{\text{avg}} \times 55,
\]

\[
E_{\text{RX-GBP}} = (N_{\text{RX-qry-msg}} \text{len}_{\text{query-msg}} + N_{\text{RX-rpl-msg}} \text{len}_{\text{reply-msg}} + N_{\text{RX-rpt-msg-gbp}} \text{len}_{\text{rpt-msg-gbp}}) E_{\text{RX}} = E_{\text{RX}} \times (L + M) \times 87 + E_{\text{RX}} \times (L + M) \times \rho_{\text{node}} \times 49 + E_{\text{RX}} \times (L + M) \times \text{hop}_{\text{avg}} \times 55.
\]

In ISS, communication overhead includes broadcasting and receiving event-trigger-signal overhead, transmitting and receiving token-pass-msg overhead, and sending and receiving event report message (rpt-msg) overhead. (1) In the transmitting event-trigger-signal phase, tag-nodes and token nodes send event-trigger-signals and their neighbor nodes receive event-trigger-signals, which is similar to the corresponding phases in the GBP scheme. In the sending rpt-msgs phase, since every node which receives the event-trigger-signal generates a rpt-msg and delivers it toward the sink node, the number of token-pass-msgs transmitted \( N_{\text{TX-tok-pass-msg-iss}} \) and \( N_{\text{TX-tok-pass-msg-ss}} \) are \( L + M \) and \( L + M \rho_{\text{node}} \), respectively. (2) In the transmitting token-pass-msg phase, token nodes, their neighboring nodes, and tag-nodes’ neighbor nodes transmit token-pass-msgs. Thus, the number of token-pass-msgs transmitted \( N_{\text{TX-token-msg-iss}} \) and the number of token-pass-msgs received \( N_{\text{RX-token-msg-iss}} \) are \( L + M \) and \( L + M \rho_{\text{node}} \), respectively. (3) In the sending rpt-msgs phase, the chosen reporter nodes generate rpt-msgs and deliver them to the sink node. Since one tag-node or one token node triggers a rpt-msg, similar to the sending rpt-msgs phase in GBP, we have \( N_{\text{TX-rpt-msg-iss}} = N_{\text{RX-rpt-msg-iss}} \) and \( N_{\text{TX-rpt-msg-gbp}} = (L + M) \rho_{\text{node}} \). Table 3 shows the total number of messages transmitted and received in each phase in the ISS scheme. Let \( E_{\text{TX-ISS}} \) and \( E_{\text{RX-ISS}} \) be the energy consumption for transmitting and receiving messages in the ISS scheme, respectively. Then, we have

\[
E_{\text{TX-ISS}} = (N_{\text{TX-trig-msg-iss}} \text{len}_{\text{trig-msg}} + N_{\text{TX-token-msg-iss}} \text{len}_{\text{token-pass-msg-iss}} + N_{\text{TX-rpt-msg-iss}} \text{len}_{\text{rpt-msg-iss/iss}}) E_{\text{TX}} = E_{\text{TX}} \times (L + M) \times 17 + E_{\text{TX}} \times (L + M) \times \rho_{\text{node}} \times 55 + E_{\text{TX}} \times (L + M) \times \text{hop}_{\text{avg}} \times 39,
\]

\[
E_{\text{RX-ISS}} = (N_{\text{RX-trig-msg-iss}} \text{len}_{\text{trig-msg}} + N_{\text{RX-token-msg-iss}} \text{len}_{\text{token-pass-msg-iss}} + N_{\text{RX-rpt-msg-iss}} \text{len}_{\text{rpt-msg-iss/iss}}) E_{\text{RX}} = E_{\text{RX}} \times (L + M) \times 17 + E_{\text{RX}} \times (L + M) \times \rho_{\text{node}} \times 55 + E_{\text{RX}} \times (L + M) \times \text{hop}_{\text{avg}} \times 39.
\]

In ISS, communication overhead includes broadcasting and receiving event-trigger-signal overhead, sending and receiving event report message (rpt-msg) overhead, and transmitting and receiving token-pass-msg overhead. In both transmitting event-trigger-signal phase and sending token-pass-msg phase, it is similar to the corresponding phases in the ISS scheme. In the sending rpt-msgs phase, since every node which receives the event-trigger-signal generates a rpt-msg and delivers it toward the sink node, the number of token-pass-msgs transmitted \( N_{\text{TX-token-pass-msg-iss}} \) is \( (L + M) \rho_{\text{node}} \). Table 4 shows the total number of messages transmitted and received in each phase in the SS scheme. Let \( E_{\text{TX-SS}} \) and \( E_{\text{RX-SS}} \) be the energy consumption for transmitting and receiving messages in the SS scheme, respectively. Then, we have

\[
E_{\text{TX-SS}} = (N_{\text{TX-trig-msg-ss}} \text{len}_{\text{trig-msg}} + N_{\text{TX-token-msg-ss}} \text{len}_{\text{token-pass-msg-ss}} + N_{\text{TX-rpt-msg-ss}} \text{len}_{\text{rpt-msg-iss/iss}}) E_{\text{TX}} = E_{\text{TX}} \times (L + M) \times 17 + E_{\text{TX}} \times (L + M) \times \rho_{\text{node}} \times 35 + E_{\text{TX}} \times (L + M) \times \text{hop}_{\text{avg}} \times 39,
\]
Table 2: Total number of messages transmitted and received of each type in the GBP scheme.

| Type          | Tx         | Rx                  |
|--------------|------------|---------------------|
| query-msg    | \(n_{\text{row}} \times n_{\text{clm}}\) | \(L + M\)          |
| reply-msg    | \(L + M\)  | \((L + M)\rho_{\text{node}}\) |
| rpt-msg      | \((L + M)\text{hop}_{\text{avg}}\) | \((L + M)\text{hop}_{\text{avg}}\) |

Table 3: Total number of messages transmitted and received of each type in the ISS scheme.

| Type          | Tx         | Rx                  |
|--------------|------------|---------------------|
| event-trigger-signal | \(L + M\)  | \((L + M)\rho_{\text{node}}\) |
| token-pass-msg | \((L + M)\rho_{\text{node}}\) | \((L + M)\rho_{\text{node}}\) |
| rpt-msg      | \((L + M)\text{hop}_{\text{avg}}\) | \((L + M)\text{hop}_{\text{avg}}\) |

Table 4: Total number of messages transmitted and received of each type in the SS scheme.

| Type          | Tx         | Rx                  |
|--------------|------------|---------------------|
| event-trigger-signal | \(L + M\)  | \((L + M)\rho_{\text{node}}\) |
| token-pass-msg | \((L + M)\rho_{\text{node}}\) | \((L + M)\rho_{\text{node}}\) |
| rpt-msg      | \((L + M)\text{hop}_{\text{avg}}\) | \((L + M)\text{hop}_{\text{avg}}\) |

Similarly, according to (30), (32), and (34), we can derive

\[
E_{\text{Rx-GBP}} - E_{\text{Rx-ISS}} = E_{\text{Rx}} \times \left(87 + 32\rho_{\text{node}} + 16\text{hop}_{\text{avg}} - 55\rho_{\text{node}}^2\right),
\]

\[
E_{\text{Rx-GBP}} - E_{\text{Rx-SS}} = E_{\text{Rx}} \times \left(87 + 32\rho_{\text{node}} + 55\text{hop}_{\text{avg}} - 35\rho_{\text{node}}^2 - 39\rho_{\text{node}}\text{hop}_{\text{avg}}\right),
\]

\[
E_{\text{Rx-ISS}} - E_{\text{Rx-SS}} = E_{\text{Rx}} \times \left(20\rho_{\text{node}}^2 + 39\text{hop}_{\text{avg}} - 39\rho_{\text{node}}\text{hop}_{\text{avg}}\right).
\]

In (38), if

\[
\rho_{\text{node}} \geq \frac{32 + \sqrt{1024 + 220(87 + 16\text{hop}_{\text{avg}})}}{110},
\]

then \(E_{\text{Rx-GBP}} \leq E_{\text{Rx-ISS}}\); that is, the energy consumed to receive messages in GBP will be less than that in ISS. For example, let \(\text{hop}_{\text{avg}} = 20\); thus, if \(\rho_{\text{node}} \geq 4\), then the energy consumed to receive messages in GBP is less than that in ISS. Obviously, in most cases, as WSNs are densely deployed, we have \(E_{\text{Rx-GBP}} \leq E_{\text{Rx-ISS}}\).

In (39), if

\[
\rho_{\text{node}} \geq \frac{-\tilde{b} + \sqrt{\tilde{b}^2 + 140(87 + 55\text{hop}_{\text{avg}})}}{70},
\]

where \(\tilde{b} = (39\text{hop}_{\text{avg}} - 32)\), then \(E_{\text{Rx-GBP}} \leq E_{\text{Rx-SS}}\); that is, the energy consumed to receive messages in GBP will be less than that in SS. For example, let \(\text{hop}_{\text{avg}} = 20\); thus, if \(\rho_{\text{node}} \geq 2\), then the energy consumed to receive messages in GBP is less than that in SS. Obviously, in most cases, as WSNs are densely deployed, we have \(E_{\text{Rx-GBP}} \leq E_{\text{Rx-SS}}\).

In (40), if

\[
\rho_{\text{node}} \geq \frac{39\text{hop}_{\text{avg}} - \sqrt{(39\text{hop}_{\text{avg}})^2 - 3120\text{hop}_{\text{avg}}}}{40},
\]

with \(\text{hop}_{\text{avg}} \geq 3\), then \(E_{\text{Rx-ISS}} \leq E_{\text{Rx-SS}}\). Otherwise, \(E_{\text{Rx-ISS}} \geq E_{\text{Rx-SS}}\).
6.1.3. Computation Overhead. Compared with the SS scheme and the ISS scheme, the most time-consuming task for computation in the GBP scheme is the scalar point multiplication operation. Similar to [9, 10, 41, 42], we ignore conventional hash operations and symmetric-key encryption/decryption operations as they consume much less energy than scalar point multiplications and transmitting/receiving operations. According to [43, 44], the energy consumption of the ECC-160 scalar point multiplication on TelosB sensor nodes with the 16-bit MSP430 microcontroller running at 4 MHz is approximately 17 mJ. As reported in [44], a CC2420 transceiver with a claimed data rate of 250 kbps used in TelosB nodes consumes 5.76 and 6.48 $\mu$J to transmit and receive one byte, respectively. As mentioned above, the byte length of a query-msg and a reply-msg in GBP are 87 and 49 bytes, respectively. Thus, the energy cost for receiving a query-msg and replying a reply-msg by a tag-node is approximately 0.846 mJ. Let $E_{rx-auth-tx}$ denote the total energy consumption for receiving one query-msg, authenticating one query-msg, and replying one reply-msg. Thus, we have $E_{rx-auth-tx} = 17.846$ mJ (i.e., 17 + 0.846). Assume that on average an object (a tag-node) in the network monitors the query-msg every 1 minute. Then, the energy consumption is $24 \times 60 \times 17.846 = 25,699$ J for a tag-node per day. Since sensor nodes are usually powered by 2 AA batteries with 18720 J [45], so a tag-node could work about 2 years with GBP. Therefore, the computation cost and the communication cost of GBP are acceptable for sensor nodes.

6.2. Simulation Results. In this section, we present the simulation results of our methods and the SS method in terms of communication overhead. For the convenience of comparison, we assume that the event-trigger-signals emitted by token nodes and tag-nodes are distinguishable for reporter sources in SS.

6.2.1. Simulation Setup. The simulation is based on the Castalia simulator [46]. In the simulation, 6000 sensor nodes are randomly generated and distributed in a 1000 m x 1000 m area. The sink node is located at the center of the field. During the simulation, we assume that there is only one monitored object in the network as [9, 10]; that is, there is only one tag-node in the network. Multiple fake monitored objects, that is, token nodes, are randomly chosen from the sensor nodes and simulated in the field. For each node, including the tag-node, the transmission range is 50 m. Thus, on average, each node has 47 neighbors. As [9, 10], we focus our simulation evaluation on how much communication cost we have to pay to achieve a given level of location privacy. For each experiment, there is only one event generated by each tag-node or by each token node. We repeated the experiment 20 times with different network topologies and all the results were obtained by computing the average of all corresponding results.

6.2.2. Simulation Results. Figures 6 and 7 show the communication costs for transmitting and receiving messages of the network at different privacy levels, respectively. Figure 8 shows the total communication costs in the network at different privacy levels.

Figure 6 shows that, (1) with the privacy level increasing, the energy consumed to transmit messages in GBP is less than that in both ISS and SS; (2) the energy consumed to transmit messages in ISS is less than that in SS, which are in accord with the analysis in Section 6.1.2. Figure 7 shows that (1) the energy consumed to receive messages in GBP is less than that in both ISS and SS; (2) the energy consumed to receive messages in SS is less than that in ISS. The reason is that $p_{node} = 47 > hop_{avg}$. The main communication cost for receiving messages in both ISS and SS is the part for receiving token-pass msgs and that is proportional to $O(p_{node}^2)$. Figure 8 shows that the communication overhead of GBP is much lower than that of both SS and ISS with the privacy requirement increasing. Although the total communication cost of ISS is higher than that of SS on the whole, the communication cost of ISS is comparable to that of SS under low privacy requirement and it can let a reporter source generate effective event reports while the SS cannot.
From Figure 6 to Figure 8, we also see that in the three schemes the communication overhead increases with the privacy requirement. However, the communication overhead of GBP demonstrates a relatively slower increase compared to the SS scheme and the ISS scheme. Note that when we also consider the computation overhead, as mentioned earlier, we only need to add the additional computation overhead of GBP to our results (i.e., 17 m), the scalar point multiplication operation of the tag-nodes). However, from Figure 6 to Figure 8, we can see that it would not have much influence on the results.

In summary, both the ISS method and the GBP method are good choices to provide the data source location privacy service. Moreover, the GBP scheme is the better choice with lower energy cost.

### 7. Conclusions

Prior work on protecting the DSLP against a global eavesdropper was based on the panda-hunter game model (PHGM) and rarely considered the communication between data sources and reporter sources, which restricts their applications and even causes their methods to fail. Only the SS method considered the event trigger process from monitored objects. However, the SS method still has two limitations. First, the reporter source cannot generate effective event reports as the event-trigger-signals emitted by token nodes and tag-nodes are indistinguishable, which was ignored in SS. Second, it is unsuitable to track multiobjects accurately. In order to solve the first problem of SS, an improved source simulation (ISS) method was proposed by adjusting the event report strategy under the PHGM. To overcome the disadvantage of the PHGM, we proposed an updated-panda-hunter game model (UPHGM) and gave the formal model of the DSLP issues. And an energy-efficient grid-based pull (GBP) scheme was presented to protect the DSLP under the UPHGM as well as to provide security services including confidentiality, authentication, integrity, and freshness.

The ISS scheme uses the token pass message to distinguish different events and only one node will generate event report message for each event-trigger-signal; hence, its communication overhead for transmitting messages is lower than that of SS, which causes the neighbor nodes of each event-trigger-signal to generate event reports. Although the total communication cost of ISS is higher than that of SS, the communication cost of ISS is comparable to that of SS under low privacy requirement and it allows a reporter source to generate effective event reports while the SS scheme cannot. The GBP scheme adopts the pull scheme, IBC-based technique, and pseudonym technique to provide the authentication service and to reduce storage overhead of key material. In order to reduce the communication overhead of GBP, we presented the optimal grid size to meet connectivity requirement and query requirement. Both the theoretical analysis and the simulation results show that GBP outperforms both ISS and SS in terms of energy cost on the whole.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant no. 60873199. The authors are grateful to the anonymous reviewers for their insightful comments.

### References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” IEEE Communications Magazine, vol. 40, no. 8, pp. 102–105, 2002.

[2] T. Arampatzis, J. Lygeros, and S. Manesis, “A survey of applications of wireless sensors and wireless sensor networks,” in Proceedings of the 20th IEEE International Symposium on Intelligent Control and the 13th Mediterranean Conference on Control and Automation (ISIC ‘05 and MED ‘05), pp. 719–724, Limassol, Cyprus, June 2005.

[3] C. F. Garcia-Hernandez, P. H. Ibarguengoytia-Gonzalez, J. Garcia-Hernandez, and J. A. Perez-Diaz, “Wireless sensor networks and applications: a survey,” International Journal of Computer Science and Network Security, vol. 7, no. 3, pp. 264–273, 2007.

[4] Y. Zhou, Y. Fang, and Y. Zhang, “Securing wireless sensor networks: a survey,” IEEE Communications Surveys and Tutorials, vol. 10, no. 3, pp. 6–28, 2008.

[5] X. Q. Chen, K. Makki, K. Yen, and N. Pissinou, “Sensor network security: a survey,” IEEE Communications Surveys and Tutorials, vol. 11, no. 2, pp. 52–73, 2009.

[6] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, “Enhancing source-location privacy in sensor network routing,” in Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS’05), pp. 599–608, Columbus, Ohio, USA, June 2005.

[7] R. Rios and J. Lopez, “Analysis of location privacy solutions in wireless sensor networks,” IET Communications, vol. 5, no. 17, pp. 2518–2532, 2011.
[8] N. Li, N. Zhang, S. K. Das, and B. Thuraisingham, “Privacy preservation in wireless sensor networks: a state-of-the-art survey,” Ad Hoc Networks, vol. 7, no. 8, pp. 1501–1514, 2009.

[9] K. Mehta, D. Liu, and M. Wright, “Location privacy in sensor networks against a global eavesdropper,” in Proceedings of the 15th IEEE International Conference on Network Protocols (ICNP ’07), pp. 314–323, Beijing, China, October 2007.

[10] K. Mehta, D. Liu, and M. Wright, “Protecting location privacy in sensor networks against a global eavesdropper,” IEEE Transactions on Mobile Computing, vol. 11, no. 2, pp. 320–336, 2012.

[11] M. Shao, Y. Yang, S. Zhu, and G. Cao, “Towards statistically strong source anonymity for sensor networks,” in Proceedings of the IEEE Communications Society Conference on Computer Communications (INFOCOM ’08), pp. 466–474, Phoenix, Ariz, USA, April 2008.

[12] A. Abbasi, A. Khonsari, and M. S. Talebi, “Source location anonymity for sensor networks,” in Proceedings of the 6th IEEE Consumer Communications and Networking Conference (CCCN’09), pp. 1–5, Las Vegas, Nev, USA, January 2009.

[13] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran, “Statistical framework for source anonymity in sensor networks,” in Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM ’10), pp. 1–6, Miami, Fla, USA, December 2010.

[14] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran, “Towards a statistical framework for source anonymity in sensor networks,” IEEE Transactions on Mobile Computing, vol. 12, no. 2, pp. 248–260, 2013.

[15] Y. Yang, M. Shao, S. Zhu, B. Uraganar, and G. Cao, “Towards event source unobservability with minimum network traffic in sensor networks,” in Proceedings of the 1st ACM Conference on Wireless Network Security, pp. 77–88, Alexandria, Va, USA, March 2008.

[16] K. Bicakci, H. Gualtekin, B. Tavli, and I. E. Bagci, “Maximizing lifetime of event-unobservable wireless sensor networks,” Computer Standards and Interfaces, vol. 33, no. 4, pp. 401–410, 2011.

[17] W. Yang and W. Zhu, “Protecting source location privacy in wireless sensor networks with data aggregation,” in Proceedings of the International Conference on Ubiquitous Intelligence and Computing (UIC ’10), pp. 252–266, Xian, China, October 2010.

[18] Y. Ouyang, Z. Le, D. Liu, J. Ford, and F. Makedon, “Source location privacy against laptop-class attacks in sensor networks,” in Proceedings of the 4th International Conference on Security and Privacy in Communication Networks (SecureComm ’08), Istanbul, Turkey, September 2008.

[19] S. Kokali-Filipovic, F. Le Fessant, and P. Spasojevic, “The quality of source location protection in globally attacked sensor networks,” in Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops ’11), pp. 44–49, Seattle, Wash, USA, March 2011.

[20] C. Ozturk, Y. Zhang, and W. Trappe, “Source-location privacy in energy-constrained sensor network routing,” in Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN ’04), pp. 88–93, Washington, DC, USA, October 2004.

[21] Y. Xi, L. Schwiebert, and W. Shi, “Preserving source location privacy in monitoring-based wireless sensor networks,” in Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS ’06), pp. 1–8, 2006.

[22] W. Wang, L. Chen, and J. Wang, “A source-location privacy protocol in WSN based on locational angle,” in Proceedings of the IEEE International Conference on Communications (ICC ’08), pp. 1630–1634, Beijing, China, May 2008.

[23] J. Yao and G. Wen, “Preserving source-location privacy in energy-constrained wireless sensor networks,” in Proceedings of the 28th International Conference on Distributed Computing Systems Workshops (ICDCS Workshops ’08), pp. 412–416, Beijing, China, June 2008.

[24] H. Wang, B. Sheng, and Q. Li, “Privacy-aware routing in sensor networks,” Computer Networks, vol. 53, no. 9, pp. 1512–1529, 2009.

[25] Y. Li and J. Ren, “Providing source-location privacy in wireless sensor networks,” in Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications (WASA ’09), pp. 338–347, Boston, Mass, USA, August 2009.

[26] Y. Li and J. Ren, “Preserving source-location privacy in wireless sensor networks,” in Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON ’09), pp. 1–9, Rome, Italy, June 2009.

[27] Y. Li, J. Ren, and J. Wu, “Quantitative measurement and design of source-location privacy schemes for wireless sensor networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 7, pp. 1302–1311, 2012.

[28] O. Yi, L. Zhengyi, C. Guanjing, J. Ford, and F. Makedon, “Entrapping adversaries for source protection in sensor networks,” in Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM ’06), pp. 23–32, Buffalo-Niagara Falls, NY, USA, June 2006.

[29] M. M. E. A. Mahmoud and X. Shen, “A cloud-based scheme for protecting source-location privacy against hotspot-locating attack in wireless sensor networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 10, pp. 1805–1818, 2012.

[30] W. Trappe and L. C. Washington, Introduction to Cryptography with Coding Theory, Prentice Hall, New York, NY, USA, 2002.

[31] Y. Xu, J. Heidemann, and D. Estrin, “Geography-informed energy conservation for ad hoc routing,” in Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MOBICOM ’01), pp. 70–84, Rome, Italy, July 2001.

[32] Digital Hash Standard, Federal Information Processing Standards, Federal Information Processing Standards Publication 180-1, 1995.

[33] Y. Zhang, W. Liu, Y. Fang, and D. Wu, “Secure localization and authentication in ultra-wideband sensor networks,” IEEE Journal on Selected Areas in Communications, vol. 24, no. 4, pp. 829–835, 2006.

[34] X. Cheng, A. Thaeler, G. Xue, and D. Chen, “TPS: a time-based positioning scheme for outdoor wireless sensor networks,” in Proceedings of Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’04), pp. 2685–2696, Hong Kong, China, March 2004.

[35] H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for sensor networks,” in Proceedings of the IEEE Computer Symposium on Research in Security and Privacy (S&P ’03), pp. 197–213, Berkeley, Calif, USA, May 2003.

[36] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” in Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS ’02), pp. 41–47, Washington, DC, USA, November 2002.

[37] D. Liu, P. Ning, and L. I. Rongfang, “Establishing pairwise keys in distributed sensor networks,” ACM Transactions on Information and System Security, vol. 8, no. 1, pp. 41–77, 2005.

[38] C. Karlof, N. Sastry, and D. Wagner, “TinySec: a link layer security architecture for wireless sensor networks,” in Proceedings
of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys ’04), pp. 162–175, Baltimore, Mass, USA, November 2004.

[39] “The IEEE 802.15.4 standard (ver. 2006),” 2006, http://standards.ieee.org/getieee802/download/802.15.4-2006.

[40] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” SIAM Journal on Computing, vol. 32, no. 3, pp. 586–615, 2003.

[41] Y. Zhang, W. Liu, W. Lou, and Y. Fang, “Location-based compromise-tolerant security mechanisms for wireless sensor networks,” IEEE Journal on Selected Areas in Communications, vol. 24, no. 2, pp. 247–260, 2006.

[42] M. Duan and J. Xu, “An efficient location-based compromise-tolerant key management scheme for sensor networks,” Information Processing Letters, vol. 111, no. 11, pp. 503–507, 2011.

[43] A. Liu and P. Ning, “TinyECC: a configurable library for elliptic curve cryptography in wireless sensor networks,” Tech. Rep. TR-2007-36, North Carolina State University, Department of Computer Science, 2007.

[44] G. De Meulenaer, F. Gosset, F. Standaert, and O. Pereira, “On the energy cost of communication and cryptography in wireless sensor networks,” in Proceedings of the 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communication (WiMob ’08), pp. 580–585, Avignon, France, October 2008.

[45] A. Boulis, Castalia User’s Manual, http://castalia.research.nicta.com.au/index.php/en/.

[46] Wireless Sensor Networks Simulator: Castalia, http://castalia.research.nicta.com.au/index.php/en/.