Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 601462, 14 pages
http://dx.doi.org/10.1155/2013/601462

Research Article

An Efficient and Lightweight Source Privacy Protecting Scheme for Sensor Networks Using Group Knowledge

Zhiqiang Ruan,[1] Wei Liang,[2] Decai Sun,[3] Haibo Luo,[1] and Fanyong Cheng[1]

[1] *Department of Computer Science, Minjiang University, Fuzhou, Fujian 350108, China*
[2] *School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, Hunan 411201, China*
[3] *College of Information Science and Technology, Bohai University, Jinzhou, Liaoning 121013, China*

Correspondence should be addressed to Zhiqiang Ruan; rzq_911@163.com

Received 20 December 2012; Accepted 10 March 2013

Academic Editor: Yanmin Zhu

Providing source privacy is a critical security service for sensor networks. However, privacy preserving in sensor networks is a challenging task, particularly due to the limited resources of sensor nodes and the threat of node capture attack. On the other hand, existing works use either random walk path or fake packets injection, both incurring tremendous overhead. In this work, we propose a new approach, which separates the sensor nodes into groups. The source packet is randomly forwarded within and between the groups with elaborate design to ensure communication anonymity; furthermore, members of each group exchange encrypted traffic of constant packet length to make it difficult for the adversary to trace back. One salient feature of the proposed scheme is its flexibility of trading transmission for higher anonymity requirement. We analyze the ability of our proposed scheme to withstand different attacks and demonstrate its efficiency in terms of overhead and functionality when compared to existing works.

1. Introduction

Wireless Sensor Networks (WSNs) are explored to be used in many applications ranging from military strategy to civilian purposes. Although a sensor node has certain limitations like low processing capabilities and battery supplement, the smaller size and the cost of the sensor node helps to be unspectacular in the sensing area and is useful for tracking events without being recognized.

The sensor nodes usually adopt wireless communications mode when deployed in open and harsh environments, which makes data transmissions overheard by the vicinity sensor nodes or other wireless devices. Without precaution, the adversary can overhear packet transmission and trace back the packet to the source. This can lead to the leakage of source position and the time of event packet taken place. In view of a mission critical military application, any revelation of information such as time or event location can be beneficial to the adversary and costly to the network goal. Therefore, privacy of the monitored event holds great importance that requires providing privacy to source nodes.

Source privacy is usually compromised by the contextual information of a packet, but not by the actual content of the packet. Specifically, a packet consists of a payload and a header. The payload part can be encrypted with cryptography methods to prevent adversaries from learning the information carried in intercepted packets, while the header part has to be left in clear to provide multihop routing. Hence, attackers can learn packet original source and final destination from packet headers. A viable solution is allowing neighboring nodes to establish pairwise keys whereby to encrypt packet sources and destinations. However, this countermeasure may become invalid in the presence of compromised nodes. Therefore, source privacy cannot be addressed by encryption alone.

Traffic analysis [1] enables attackers to infer the network traffic pattern and real packet sources/destinations. If noticing that a few nodes often act as packet source, the adversary

may think these nodes are important and launch targeted attacks on them. Providing communication anonymity has been regarded as an effective solution against traffic analysis [2–4] because the adversary can no longer distinguish true sources from intercepted packets.

Plenty of work is guided towards applying some form of simulating the source [5–9] or performing a random walk [5, 10] to guarantee source privacy. One primary drawback of these approaches is that a large amount of overhead is incurred to simulate a source or to redirect traffic randomly. Besides, recent works only consider eavesdropping attacks; however compromised nodes are not considered as part of the threat model where an adversary can easily capture any of sensor nodes. In this work, we consider a rather strong threat model in which the adversary can compromise nodes and is able to eavesdrop over the network communication. An example of such an adversary is a laptop class attacker who has more powerful capability to execute higher strength calculation. When a node is compromised, the adversary has access to all the cryptographic information along with data packets of past communications stored at the node.

This paper aims to protect source privacy against both passive attacks (global eavesdroppers) and active attackers (node compromise attack) as well as to guarantee fundamental security requirements such as confidentiality, data integrity, authentication, and nonrepudiation.

The contribution of the paper is as follows. We present an efficient and lightweight source privacy guaranteeing mechanism (ELSP), which applies Identity-Based Cryptography (IBC) method to complement other techniques. *First*, sensor nodes are organized into pairwise disjoint groups (or sets) to hide real packet sources among crowds of nodes. In order to conceal source identity, packet sources use pseudonyms instead of their real IDs so that any other nodes cannot ascertain the initiators of received packets. Our work differs from previous works in that pseudonyms generation does not introduce a central authority. *Second*, to provide strong communication anonymity, a random packet-forwarding strategy is presented. The source no longer sends packets along the shortest path to the destination. On the contrary, it randomly selects several forwarding nodes within each group to confuse the adversary. More importantly, members of each group exchange encrypted traffic of constant packet length to make the path untraceable. Each intermediate router (referred to as *key node*), only knowing its predecessor and successor, strips off one layer of encryption, and eventually, the receiver obtains the packet in plaintext. *Finally*, ELSP provides provably strong source anonymity against different attacks. One salient feature of ELSP is its flexibility of trading transmission for higher anonymity requirement. A detailed analysis of ELSP is provided, and a comprehensive comparison with existing schemes is presented to show its effectiveness and efficiency.

The rest of this paper is organized as follows. Section 2 reviews the existing work on source privacy, and Section 3 presents the models and design goals. Next, we describe the ELSP scheme in Section 4, followed by the detailed analysis and performance evaluation in Section 5. Finally, we have the conclusions in Section 6.

2. Related Work

Recently, source privacy in sensor networks has drawn widespread concern. Kamat et al. proposed a phantom routing protocol for flooding and single path routing [5]. They were the earliest researchers to study source privacy and present multiple techniques to guarantee the objective. One technique uses fake sources with nodes sending fake packets to mislead the adversary. The other technique is called phantom routing which takes a random walk before forwarding the packet towards the base station in order to increase the cost of the adversary to backtrack to the source. Although the schemes are robust, they have a large overhead involved and can not withstand the collaborative attacks.

Mehta et al. similarly propose two schemes called per iodic data aggregation and source simulation to overcome global eavesdropping attacks [6]. The source simulation scheme is similar to the fake sources technique proposed in [5]. In per iodic aggregation scheme, each node reports back to the base station per iodically, regardless of whether it detects an event or not. The drawback of per iodic collection scheme is the latency incurred as well as overhead, while in source simulation scheme, it is the overhead introduced.

Wang et al. [10] propose a parallel-routing protocol to maximize the time for adversary traceback to source. The packets from the same source are passed through different paths to the base station. Furthermore, a weighted random stride routing is presented that breaks the entire routing into rounds. Li et al. [11] adapt the conventional function of data mules to design a new protocol for securing source location privacy, namely, the Mules-Saving-Source (MSS) protocol, which provides a-angle anonymity. Although they are nice schemes, one of the prerequisites is that the sensor location should provided for determining the forwarding angle. Moreover, they be fail in protecting source privacy in case of a global eavesdropping adversary.

In [12], the authors propose a scalable hop-by-hop authentication scheme based on elliptic curve cryptography (ECC), which enables any intermediate node to transmit an unlimited number of messages without determining the degree of threshold suffering in the polynomial-based scheme [13, 14]. They further propose a scheme to provide source location privacy through routing to a randomly selected intermediate node and a network-mixing ring [15]. Although these two schemes can provide comparable source privacy, the first scheme brings another problem of handling the selection of the AS (ambiguity set), while the second scheme again involves taking a random walk.

Nezhad et al. devise a label-switching based technique to meet source and base station anonymity [16]. One of the restrictions is the requirement of global network information with which the base station constructs a routing tree. Each link of the routing path has a separate label when a node receives a packet, it changes the label of the packet to the upstream link and the packet gets propagated. Except for the base station demanding global knowledge of the network, each node has to perform exhaustive processing and reconstruction of the packets routing over them.

Shao et al. proposed FitProbRate [8], an exponentially distributed dummy traffic generation scheme, to maintain source anonymity. This work differs from other similar works with the dummy traffic generated at a dynamic rate decided by the Fitprob parameter. It is a great improvement over source simulation and fake sources but still has the drawback of having overhead due to dummy packet generation.

There is some other literature that considers privacy issue in WSNs. Chai et al. [17] provide the sink-location privacy against a powerful adversary with a global view, but it not considered the source privacy, which is the main focus of this paper. Chow et al. [18] focused on providing both location-monitoring services and source privacy. Di Pietro and Viejo [19] addressed the problem of querying by the base station to provide the MAX of sensor-stored readings and get a trade-off between accuracy of the result and overhead. Li and Hwang [20] propose a lightweight anonymous routing protocol in secure wireless ad hoc networks. Therefore, they are orthogonal to our paper.

All the works discussed by now just consider a passive adversary. Most consider a local eavesdropping adversary with a few providing solution about global eavesdropping adversary. In ELSP, we consider an eavesdropping adversary having node compromising capabilities. We present a packet-altering scheme, which has lesser overhead compared to existing schemes. Also, when compared to a label-switching scheme as [16], ELSP does not need every node on the path to perform packet transformation and does not need the base station to be aware of the network topology.

Our work is partly inspired by Mix Nets [2] and Crowds [4]. In particular, Crowds is used for intragroup communications; each packet travels a random path whose length follows a given geometric distribution. Since each packet may traverse a group through different number of nodes, each group in fact serves as a virtual *mix* through which packets are mixed.

3. Models and Design Goals

3.1. Network Model. We consider the sensor network comprised of homogeneous sensor nodes spread over a wide area. The sensor nodes are responsible for detecting events and reporting back to the base station. The base station is assumed to be secure and has unlimited resources when compared to the general sensor nodes. The occurrence of the events can be irregular and random in nature. ELSP can be used for many applications requesting source node privacy protection. A class of applications is used to track endangered animals or birds. The existences of such species need to be preserved from hunters or poachers as they have potential market value; meanwhile, they need to be studied. In a word, we consider the sensor network deployed in a wild for sensing endangered animals (e.g., South China tiger). It is a homogeneous network with small size sensor nodes dispersed over a vast area. Sensor nodes sense their environment for the presence of endangered species and report back to the base station. Note that multiple sensor nodes can detect the event simultaneously, and they will independently report it back to the base station.

The base station collects the packets and identifies the emergence of the tracked object and studies the hunting way and their living conditions. Given the tracked object being swoop and swift, we can have multiple nodes detecting the event in a specified time, and then not having any detection for a long time. We are based on the following consideration: if the animal appears somewhere, it did so to either prey or to have a rest, and this increases the possibility of the animal haunting to that location thereby needing source location privacy for the detecting packet. The event in some applications can be sporadic, but there are still other communications between the sensor nodes, resulting in the generation of packet traffic.

3.2. Attack Model. Since the high profits are brought from animal hunting, it is no wonder that the adversaries would equip themselves with advanced equipment, which means they have overwhelming technical advantages over general sensor nodes, such as sufficient energy resource, powerful computation capability, and large storage space. Therefore, the adversaries can overhear communication at much larger distances compared to a sensor node. They also possess the ability to compromise nodes. Specifically, the adversaries can operate the following two modes.

3.2.1. Global Eavesdropping Mode (External Attacks). The adversary in this mode can carry out passive attacks, such as eavesdropping of the communications and can correlate the transmission of the packet over multiple hops. Although the adversary may not able to ascertain the contents of the packet, it has the capability to compare two packets. Note that the adversary will not interfere with the function of the network, such as destroying sensor nodes, altering the routing path, or modifying packets because such activities can be easily identified.

3.2.2. Stealth Mode (Internal Attacks). The adversaries in this mode can compromise sensor nodes and get access to all the cryptographic information stored on them. They can further decode the packet and get information as available to any rightful sensor node. The adversary can compromise sensor nodes randomly from the network or geographically close to each other (e.g., neighboring nodes).

The goal of an attacker is to acquire the location and time of event occurrence, either by passive eavesdropping or active node compromise. In the worst case, the adversary may employ both.

3.3. Design Goals. The objective of this paper can be summarized as follows.

(1) The adversary can not get the source information by analyzing the traffic pattern.

(2) The adversary can not get the source information, even though a few network nodes are compromised.

Table 1: List of used notations.

Notation	Description
S, D	Source and destination (the basestation)
g_i	The ith group
$g_{i,j}$	The network ID of the jth node in group i
G_1, G_2	The additive group and multiplicative group of order q
K_V	The private key of node V
K_{UV}	The share key between nodes U and V
α	The total number of groups the packets pass through
$\mathrm{KN}_{i,z}$	The key node of group i, its group index is z
$P_{i,j}$	The proxy node of group i
$A_{S,i}$	The pseudonym of S for the ith group
K_i	The shared key between S and key node $\mathrm{KN}_{i,z}$
K_D	The shared key between S and D
Υ_i	The ith path object
Ω	The payload of the packet
Ψ_i	The packet before entering the ith group

(3) Only the base station can distinguish the source location through the messages received. The recovery of the source information from the received message should be very efficient.

(4) The length of each message must be as short as possible to save the energy of sensor node.

3.4. Notations. For the sake of clarity and convenience for the readers, we list some major notations in Table 1 to be used throughout this paper.

4. The Proposed ELSP Scheme

In this section, we first give the basic idea of ELSP, and then elaborate on its design.

4.1. Outline of ELSP. In ELSP, sensors are divided into pairwise disjoint sets called groups. We take source S and destination D as an example to present the basic idea of ELSP. Without loss of generality, we assume that S and D are in different groups.

To secretly send packets to D, node S randomly picks one node from every group which is called a *key node* through which every packet will be routed. ELSP adopts the idea of mix-nets [2]. Specifically, S packs the message for D by several layers of encryption. The packed message is then routed through the key nodes; each of them strips off one layer of encryption and then transmits to the next key node. By constructing the packet appropriately, we can guarantee that every key node knows neither other key nodes nor how many key nodes separate it between S and D. Source S is thus hidden from all the key nodes.

ELSP integrates several techniques to thwart both internal and external attackers. For example, to cover the transmission behavior, S first sends the packets to a randomly chosen group peer instead of directly sending it to the first key node. Then, after receiving a packet from its group peer, each node will send the packet to the first group with some probability, say ρ, or randomly choose another group peer to which the packet is sent. Finally, the packet will be sent to a randomly chosen *proxy node* in each group other than the first key node, which in turn forwards the packet to the first key node. We will show that this method assists in hiding the first key node from global eavesdropper. The packet will be forwarded in subsequent groups in a similar way. Furthermore, ELSP requires the members of every group to exchange encrypted packet of constant length to prevent attackers from tracing the packet. All these measures are attempted to make it difficult for the adversary to locate packet sources with tunable communication overhead.

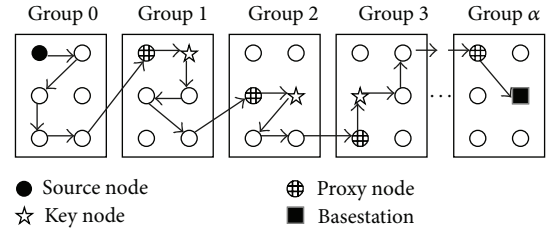


Figure 1: A diagram of ELSP.

An example is illustrated in Figure 1, where each group g_i ($1 \leq i \leq \alpha$) forms one layer. We denote by $\mathrm{KN}_{i,j}$ the key node in each group; destination D is in group α and is actually $\mathrm{KN}_{\alpha,3}$. Moreover, nodes $p_{i,j}$ in each group are proxy nodes. As we can see, source S forms a packet which is forwarded through four random nodes and then passed to $P_{1,0}$ which, in turn, routes the packet to $\mathrm{KN}_{1,1}$. Only the key node can strip off one layer of encryption, and then the modified packet passes three random nodes to reach the next group. This process continues until the packet reaches D (the base station). ELSP can ensure that each key node cannot determine which packet layer it resides at, that is, its distance from the source or destination group. This process will gradually be clear when we come back to it in Section 4.2.6.

In the following section, we will give the design details of ELSP, including group formation, group traffic maintenance, key distribution, and packet forwarding.

4.2. Detailed Description of ELSP

4.2.1. Group Formation. It is quite common that sensor nodes are deployed in groups; that is, a group of sensors are deployed at a single deployment point, and the probability distribution function (e.g., a two-dimensional Gaussian distribution) of the final resident points of all the sensors in each batch (or group) are the same [21, 22]. We assume such a group-based deployment, and we model the deployment knowledge as follows.

We consider a sensor network with N sensor nodes. Before the network deployment, the network owner divides N sensor nodes into M pairwise disjoint groups denoted by $\{g_0, \ldots, g_{M-1}\}$. The relation between sensor nodes and groups should hold the following conditions:

(1) $g_i \cap g_j = \varnothing$, for all $i, j \in \{0, \ldots, M-1\}$, $i \neq j$, $\sum_{i=0}^{M-1} |g_i| = N$.

(2) $V \in g$ (V denotes a sensor node, g denotes all groups set).

Sensor nodes in each group g_i are indexed from 0 to $|g_i| - 1$. Let $g_{i,j}$ denote the network ID of the jth node in group g_i, where $0 \leq j \leq |g_i| - 1$. Note that the groups may have different numbers of sensor nodes. Each sensor node is preloaded with the information regarding the affiliation it belongs to, the network ID of every other sensor node, and the index in that group.

4.2.2. Key Distribution and Agreement. As with other schemes, ELSP requires sensor nodes to establish appropriate cryptographic keys. In this work, we assume a key distribution scheme (e.g., [23]) based on Identity-Based Cryptography (IBC) [24]. However, ELSP can also rely on other suitable key distribution schemes by taking advantages of deployment knowledge of the deployed sensor nodes in a sensor network [25].

Since ELSP targets a single owner WSN, there exists a trusted authority (TA) to bootstrap the network. Before the network deployment, the TA selects a large prime q, a master secret key $\kappa \in z_q^*$, an additive group G_1 of order q, a multiplicative group G_2 of order q, a bilinear map $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$, and a hash function $H : \{0,1\}^* \rightarrow G_1^*$ which maps arbitrary binary strings into nonzero points in G_1. Improved Weil [24] and Tate [26] pairing are examples of the bilinear map $\hat{e}$, and we refer the readers to [24, 26] for the detailed properties of $\hat{e}$.

Each sensor node, say V, is equipped with $\langle q, G_1, G_2, \hat{e}, H \rangle$. Additionally, it has a unique public/private key pair, where the public key is the unique network ID, and the private key $K_V = \kappa H(V)$ is acquired from the TA. Note that the master secret κ cannot be linked from any public/private key pair like V/K_V [23, 24, 26], so it is only known to the TA.

In ELSP, any pair of sensor nodes, say U and V, can establish a shared key independently without communicating with each other. In particular, U and V compute

$$K_{UV} = \hat{e}\left(K_U, H(V)\right), \qquad K_{VU} = \hat{e}\left(K_V, H(U)\right). \quad (1)$$

Here, $K_{UV} = K_{VU}$ because of the symmetric and bilinear properties of $\hat{e}$ [23, 24, 26]. Furthermore, each sensor node can create many pseudonyms and the corresponding private keys. For instance, V can choose a random integer $r_V \in z_q^*$ to generate a pseudonym $r_V H(V) \in G_1$ and the corresponding private key $r_V K_V = r_V \kappa H(V)$. Since G_1 is a cyclic group of order q, multiplying $H(V)$ by r_V can cloak $H(V)$ and V [26]. In other words, it is unable to link to node V given only the pseudonym $r_V H(V)$.

4.2.3. Group Traffic Maintenance. Generally, sensor node acts as a networking repeater to route packets to and from other nodes; it seems that WSN has natural source anonymity since the adversary is unable to distinguish whether a sensor node just forwarded a given packet or has initiated it. However, this argument holds only when the outgoing traffic rate of the node is not greater than its incoming traffic rate. Otherwise, the adversary can ascertain that the node has initiated some traffic, even if he cannot determine what packets originated from that node. For example, a sensor node intended to send more packets than others indicates that it probably detects animals' activities. ELSP thus must prevent this from happening.

In ELSP, in order to hide packet sources, we insert garbage data to keep the packet length constant all the time. Furthermore, any two nodes in group i, for $0 \leq i \leq M-1$, exchange a traffic rate of λ packets/second, where λ is a public system parameter. To prevent attackers from distinguishing different packets, shared secret key should be established between any two nodes for encrypting the packet using (1).

For example, nodes U and V are both in group i which exchange packets as follows:

$$U \longleftrightarrow V : \{t \mid \text{DATA}\}_{K_{UV}},\ h_{K_{UV}}(\text{prior} - \text{data}), \quad (2)$$

where $\{\cdot\}_*$ denotes a fixed-length message encryption using the key on the subscript; K_{UV} is the shared key of U and V using (1); t is the timestamp for ensuring message freshness; $h_{K_{UV}}(\cdot)$ is a one-way keyed hash function to guarantee message authenticity. On receiving the packet, node U (or V) first checks the message authentication code. If succeed, it then decrypts message using K_{UV} and adopts the method in Section 4.2.5 to process DATA.

Eavesdropper cannot ascertain whether sensor node U initiated a data packet, as he is unable to differentiate data from each other. ELSP also guarantees that even node V cannot ascertain whether DATA was just forwarded by U or actually originated from node U; this protects U from internal attackers if any. Intergroup traffic and intragroup traffic are of the same fixed length, which can prevent attackers from inferring any useful information from packet-length changes [1].

4.2.4. Packet Construction. Now we describe the construction of packets. Assume that source $S \in g_0$ determines to send information *info* to destination $D \in g_\alpha$. To construct a packet, S does the following procedure.

Step 1. Select one key node for each group. To complete this, node S has to rely on the normal routing protocol (e.g., min hop routing) to find the node IDs on the forwarding path between S and D. Fortunately, this can be done in the network initialization with the base station simply broadcasting a message to the whole network. The base station adds two fields to the header of this message, which are "node-in-route" (NIR) field and group ID (GID) field. Initially, these two fields are empty. Starting from the base station, whenever a node propagates the message to the next hop, the node ID and group ID of the upstream node are appended to the NIR and GID. Nodes included in NIR are excluded from the random pick at the next hop. This nonrepetitive propagation terminates until reaching every node. Finally, each node has a

path node ID list (NIDL) and the corresponding group ID list (GIDL) between itself and the base station. Node S randomly picks one node from its NIDL for each group as the key node through which every packet will be routed.

Step 2. Choose a unique random integer $r_i \in z_q^*$ for each group i whereby to calculate a pseudonym $A_{S,i} = r_i H(S) \in G_1$ and the corresponding private key $r_i K_S = r_i H(S) = \kappa A_{S,i}$. $A_{S,i}$ is used to cloak source S from key nodes at group i, as will be shown soon.

Step 3. Compute a shared key with each key node $\mathrm{KN}_{i,z}$, $i \in [1, \alpha]$, as $K_i = \hat{e}(r_i K_S, H(\mathrm{KN}_{i,z})) = \hat{e}(r_i \kappa H(S), H(\mathrm{KN}_{i,z}))$, which is used to add (by S) or peel off (by $\mathrm{KN}_{i,z}$) one layer of encryption.

We use the following approach to prevent each key node from knowing its distance from source S or destination D. Assume that destination D is in layer l, $l \in [1, \alpha]$. Source S computes a shared key with D as $K_D = \hat{e}(r_d K_S, H(D))$ and then calculates

$$\Omega = \begin{cases} \{\mathrm{info}\}_{K_D} & l = 1, \\ \left\{\{\mathrm{info}\}_{K_D}\right\}_{K_1} & l = 2, \\ \left\{\left\{\cdots\left\{\{\mathrm{info}\}_{k_D}\right\}_{k_{l-1}}\cdots\right\}_{k_2}\right\}_{k_1}, & 2 < l \le \alpha. \end{cases} \tag{3}$$

Source S then derives

$$\Upsilon_i = \begin{cases} \mathrm{PLD}_1 & i = 1, \\ \{\mathrm{PLD}_2\}_{K_1} & i = 2, \\ \left\{\cdots\left\{\{\mathrm{PLD}_i\}_{K_{i-1}}\right\}_{K_{i-2}}\cdots\right\}_{k_1}, & 3 \le i \le \alpha, \end{cases} \tag{4}$$

where $\mathrm{PLD}_i = \mathrm{Mark} \mid A_{S,i} \mid \mathrm{KN}_{i,z} \mid x$. Here, Υ_i is called a *path object* which indicates the packet path. Mark is a predetermined string that explains the legitimacy of the PLD_i. Since $\{\cdot\}_*$ is a fixed-length cipher, we have $|\Upsilon_1| = |\Upsilon_2| = \cdots |\Upsilon_\alpha|$. Finally, S constructs the packet as

$$\Psi_1 = \langle \Omega, \Upsilon_\alpha, \Upsilon_{\alpha-1}, \ldots, \Upsilon_2, \Upsilon_1 \rangle. \tag{5}$$

Suppose that S has another message for the same destination D; it can generate a new packet by just replacing the message part Ω. That is, the same set of key nodes can be used in multiple messages between S and D. However, S has to change the set of key nodes per iodically if it has many messages for D to prevent the predecessor attack [27]. Besides, different event packets from the same source should have different pseudonyms in case the two packets are correlated by the adversary.

4.2.5. Packet Forwarding and Processing. Packet Ψ_1 travels a random path in source group g_0 before entering g_1. Specifically, source S randomly selects a node from g_0 and sends Ψ_1 to it in a standard packet length. When the chosen node receives Ψ_1, it forwards Ψ_1 with probability $\rho \in [0, 1]$ to a random node in g_0 and with probability $1 - \rho$ directly to g_1. Here, ρ is a system parameter called the rabbling probability. In particular, the preloaded one-way hash function is denoted as $H(x)$, where x is the hash seed. The mapping function is $f_\rho(y)$ with $y = H(x)$. For the output range of the hash function, the mapping functions map to 1 with probability ρ and to 0 with probability $1 - \rho$. The benefit of using two functions in the system is that it gives the base station a simple way to change the rabbling probability by just updating the mapping function via a broadcast. When a node receives a packet, it calculates $f_\rho(H(x))$; if the computation maps to 1, it forwards the packet to the next group. Otherwise, it forwards to another group peer. Such random-forwarding mechanism helps to prevent the attacker from identifying source S whose effectiveness has recently been discussed in [28]. Specifically, every packet forwarder (except S) cannot tell whether the group peer-sending Ψ_1 is the packet source or just a packet forwarder.

Once a node in g_0, say F, decides to forward Ψ_1 to g_1, F picks a random node $P_{1,j}$ in g_1, $j \in [0, |g_1| - 1]$ to which the following packet modified from Ψ_1 is sent:

$$F \longrightarrow P_{1,j} : \Psi_1 = \langle \Omega, \Upsilon_\alpha, \Upsilon_{\alpha-1}, \ldots, \Upsilon_2, \mathrm{Mark} \mid A_{S,1} \mid \mathrm{KN}_{1,z} \mid x \rangle, \tag{6}$$

it is possible that $P_{1,j}$ itself happens to be $\mathrm{KN}_{1,z}$ in which case $j = z$. Note that Ψ_1 should be sent in an encrypted packet to $P_{1,j}$ which then directly forwards it to $\mathrm{KN}_{1,z}$ after verifying Ψ_1. Since eavesdroppers cannot distinguish this packet from others between $\mathrm{KN}_{1,z}$ and $P_{1,j}$, they cannot immediately determine that $\mathrm{KN}_{1,z}$ is the key node in g_1. If $P_{1,j}$ is a compromised node, then the attackers confirm $\mathrm{KN}_{1,z}$ as the key node. But they still cannot ascertain whether $\mathrm{KN}_{1,z}$ is the packet destination or help them to identify the packet source. On receiving Ψ_1, $\mathrm{KN}_{1,z}$ assumes that it was originated from node $A_{S,i}$ with which to compute a shared key as $K_{1,z} = \hat{e}(K_{\mathrm{KN}_{1,z}}, A_{S,1})$. There are four cases.

(i) If $\mathrm{KN}_{1,z}$ is a key node, then $K_{1,z} = \hat{e}(K_{\mathrm{KN}_{1,z}}, A_{S,1}) = \hat{e}(sK_{\mathrm{KN}_{1,z}}, r_1 H(S))$ which is equivalent to K_1 [23, 24, 26].

(ii) If $\mathrm{KN}_{1,z}$ is destination D, then $K_{1,z} = \hat{e}(K_D, A_{S,1}) = \hat{e}(sH(D), r_1 H(S))$ which is equivalent to K_D [23, 24, 26].

(iii) If the above conditions are all met, then we have $K_1 = K_D = K_{1,z}$.

(iv) Otherwise, $K_{1,z}$ is neither equal to K_1 nor K_D.

Then it attempts using $K_{1,z}$ to decrypt Ω in Ψ_1; if the decryption result has a predefined message format, then $\mathrm{KN}_{1,z}$ realizes itself as the destination, or else, the decryption result can be ignored. Let $\mathrm{msg}^{x:y}$ denote the output of decrypting msg using shared keys $K_x, K_{x+1}, \ldots, K_{y-1}, K_y$ sequentially, which is executed by nodes $\mathrm{KN}_{x,z}, \mathrm{KN}_{x+1,z}, \ldots, \mathrm{KN}_{y-1,z}, \mathrm{KN}_{y,z}$ sequentially. For

example, $\Omega^{1:1}$ and $\Omega^{1:2}$ are separately the outputs of decrypting Ω using the shared key K_1 by $\mathrm{KN}_{1,z}$, and using K_1 and K_2 by $\mathrm{KN}_{1,z}$ and $\mathrm{KN}_{2,z}$ sequentially. Based on (3), we have

$$\Omega^{1:1} = \begin{cases} \text{random string} & l = 1, \mathrm{KN}_{1,z} \neq D, \\ \text{info} & l = 1, \mathrm{KN}_{1,z} = D, \\ \{\text{info}\}_{K_D} & l = 2, \\ \left\{\cdots\left\{\{\text{info}\}_{K_D}\right\}_{K_{l-1}}\cdots\right\}_{K_2} & 2 < l \leq \alpha. \end{cases} \tag{7}$$

Furthermore, $\mathrm{KN}_{1,z}$ uses $K_{1,z}$ to decrypt $\Upsilon_2 = \{\text{Mark} \mid A_{S,2} \mid \mathrm{KN}_{2,z} \mid x\}_{K_1}$. Since $K_{1,z} = K_1$, only $\mathrm{KN}_{1,z}$ can make a successful decryption, which knows this after seeing Mark. Node $\mathrm{KN}_{1,z}$ continues using K_1 to decrypt $\Upsilon_3, \Upsilon_4, \ldots, \Upsilon_\alpha$, and obtain $\Upsilon_3^{1:1}, \Upsilon_4^{1:1}, \ldots, \Upsilon_\alpha^{1:1}$. According to (4), it has $\Upsilon_i^{1:1} = \{\cdots\{\mathrm{PLD}_i\}_{K_{i-2}}\cdots\}_{K_2}$, for all $i \in [3, \alpha]$. Finally, $\mathrm{KN}_{1,z}$ forms a new packet

$$\Psi_2 = \left\langle \Omega^{1:1}, R_1, \Upsilon_\alpha^{1:1}, \ldots, \Upsilon_3^{1:1}, \Upsilon_2^{1:1} \right\rangle, \tag{8}$$

where $\Upsilon_2^{1:1} = \text{Mark} \mid A_{S,2} \mid \mathrm{KN}_{2,z} \mid x$, and R_1 is a random garbage data of length $|\Upsilon_1|$ used to compensate for the deletion of Υ_1 so that Υ_2 and Υ_1 are of the same format and length. Similar to Ψ_1, packet Ψ_2 takes a random path (starting from $\mathrm{KN}_{1,z}$) in g_1 before reaching g_2. Generally, each key node $\mathrm{KN}_{i,z}$, $i \in [1, \alpha - 1]$ generates packet Ψ_{i+1} in which a random string R_i is added after the message part to keep a constant packet length. R_is can not be differentiated from each other and will be treated as real path objects by subsequent key nodes. By doing so, each key node cannot identify at which layer of encryption it is.

The packet forwarding is terminated at the last key node $\mathrm{KN}_{\alpha,z}$ which receives the following packet:

$$\Psi_\alpha = \left\langle \Omega^{1:(\alpha-1)}, R_{\alpha-1}, \ldots, R_1^{2:(\alpha-1)}, \Upsilon_\alpha'^{1:(\alpha-1)} \right\rangle, \tag{9}$$

where $\Upsilon_\alpha'^{1:(\alpha-1)} = \text{Mark} \mid A_{S,\alpha} \mid \mathrm{KN}_{\alpha,z} \mid 0$, Here, x is replaced with zero of equal length by S. Then $\mathrm{KN}_{\alpha,z}$ processes Ψ_α similarly as before. $\mathrm{KN}_{\alpha,z}$ knows itself as the key node after decrypting $\Upsilon_\alpha'^{1:(\alpha-1)}$ and finding Mark there. Then it uses K_α to decrypt Ω in Ψ_α. Since $K_\alpha = K_D$, $\mathrm{KN}_{\alpha,z}$ knows that it is the message destination because the decryption result info has a predefined message format.

Since the same set of key nodes is used in delivering multiple messages from S to D, this can significantly reduce the computation overhead. Specifically, each $\mathrm{KN}_{i,z}$ knows whether it is the destination or a key node after processing the first message and can cache the corresponding source pseudonym $A_{S,i}$.

4.2.6. An Example. In order to have a better understanding with ELSP, we take the same example in Figure 1, where $\alpha = 3$ and the base station as the destination D which is actually the key node $\mathrm{KN}_{3,2}$ in g_3. For simplicity, we assume that all groups have the same size of 6.

Source S forms $\Psi_1 = \langle \Omega, \Upsilon_3, \Upsilon_2, \Upsilon_1 \rangle$, where

$$\begin{aligned} \Omega &= \left\{\{\text{info}\}_{K_D}\right\}_{K_1}, \\ \Upsilon_3 &= \left\{\{\text{Mark} \mid A_{S,3} \mid \mathrm{KN}_{3,2} \mid x\}_{K_2}\right\}_{K_1}, \\ \Upsilon_2 &= \{\text{Mark} \mid A_{S,2} \mid \mathrm{KN}_{2,3} \mid x\}_{K_1}, \\ \Upsilon_1 &= \text{Mark} \mid A_{S,1} \mid \mathrm{KN}_{1,1} \mid x. \end{aligned} \tag{10}$$

Ψ_1 passes through four nodes before leaving group g_0. Node $\mathrm{KN}_{1,1}$ receives from its proxy node $P_{1,0}$ the packet $\Psi_1 = \langle \Omega, \Upsilon_3, \Upsilon_2, \text{Mark} \mid A_{S,1} \mid \mathrm{KN}_{1,1} \mid x \rangle$ and calculates a shared key $K_{1,1}$ based on the source pseudonym $A_{S,1}$. Obviously, only $K_{1,1} = K_1$ with which $\mathrm{KN}_{1,1}$ generates $\Psi_2 = \langle \Omega^{1:1}, R_1, \Upsilon_3^{1:1}, \Upsilon_2^{1:1} \rangle$, where

$$\begin{aligned} \Omega^{1:1} &= \{\text{info}\}_{K_D}, \\ \Upsilon_3^{1:1} &= \{\text{Mark} \mid A_{S,3} \mid \mathrm{KN}_{3,2} \mid x\}_{K_2}, \\ \Upsilon_2^{1:1} &= \text{Mark} \mid A_{S,2} \mid \mathrm{KN}_{2,3} \mid x. \end{aligned} \tag{11}$$

Node $\mathrm{KN}_{1,1}$ knows itself not the message destination after decrypting Ω with $K_{1,1}$ and not finding Mark there. Because $K_{1,1} \neq K_D$, the decryption result is a random string which does not provide a predefined message format. So, $\mathrm{KN}_{1,1}$ uses mapping function $f_\rho(H(x))$ to generate the next forwarding node for Ψ_2. As we can see from Figure 1, Ψ_2 passes through three nodes before leaving g_1.

Node $\mathrm{KN}_{2,3}$ receives from the proxy node $P_{2,2}$ a packet $\Psi_2 = \langle \Omega^{1:1}, R_1, \Upsilon_3^{1:1}, \text{Mark} \mid A_{S,2} \mid \mathrm{KN}_{2,3} \mid x \rangle$ and then calculates a shared key $K_{2,3}$ based on $A_{S,2}$. This time we have $K_{2,3} = K_2$, $\mathrm{KN}_{2,3}$ then uses $K_{2,3}$ to generate $\Psi_3 = \langle \Omega^{1:2}, R_2, R_1^{2:2}, \Upsilon_3^{1:2} \rangle$, where $\Upsilon_3^{1:2} = \text{Mark} \mid A_{S,3} \mid \mathrm{KN}_{3,2} \mid x$. Again, $\mathrm{KN}_{2,3}$ knows that itself not the destination after decrypting $\Upsilon_3^{1:1}$ with $\Omega^{1:1}$.

Ψ_3 goes by two nodes before leaving g_2. Node $\mathrm{KN}_{3,2}$ receives from its proxy node $P_{3,4}$ a packet $\Psi_3 = \langle \Omega^{1:2}, R_2, R_1^{2:2}, \text{Mark} \mid A_{3,2} \mid \mathrm{KN}_{3,2} \mid 0 \rangle$ and then calculates $K_{3,2}$ to decrypts $\Omega^{1:2}$. Node $\mathrm{KN}_{3,2}$ knows that it is the message destination, as the decryption result does provide a predefined message format so the packet forwarding process succeeded.

By virtue of pseudonym, each $\mathrm{KN}_{i,z}$ considers that it receives the packet Ψ_i from source $A_{S,i}$. Besides, the addition of R_i maintains constant packet length all the time. Hence, each $\mathrm{KN}_{i,z}$ cannot identify which packet layer it resides in, and it could be at any layer $i \in [1, \alpha - 1]$ with equal probability (D knows that it locates at the last layer α). This forces the attackers to consider all packets when tracking back, and it cannot discount any packet analysis requirement upon the packet length. Using garbage filler slightly increases the overhead, but it is still very less compared to existing random walk or fake packet generation schemes as shown in Section 5.

5. Analysis and Performance Evaluation

In this section, we first give the communication and computation overhead of ELSP, and then its security about source anonymity. Finally, we conduct the simulation to demonstrate the efficiency of ELSP when compared with existing works.

5.1. Overhead Analysis

5.1.1. Communication Cost. In ELSP, a fixed packet size is used to prevent the attackers from inferring any useful information from packet-length changes. Each packet contains a constant length message part and α path objects of equal length. If the message is not long enough, it has to be padded to keep the fixed length. For convenience, we choose $\Psi_1 = \langle \Omega, \Upsilon_\alpha, \Upsilon_{\alpha-1}, \ldots, \Upsilon_2, \Upsilon_1 \rangle$ to analyze the *packet overhead* which is defined as the ratio of nonmessage part to the packet length len. Since $|\Upsilon_1| = |\Upsilon_2| = \cdots = |\Upsilon_\alpha|$, we only need to compute $|\Upsilon_1|$, where $|\Upsilon_1| = \text{Mark} \mid A_{S,1} \mid \text{KN}_{1,z} \mid x$.

The length of $A_{S,i}$ is an element in group G_1 and in fact a point on an elliptic curve over F_p [23, 24, 26]. If the prime p is of 160 bits and other pairing parameters are properly selected, it can achieve a security level equivalent to that of 1024-bit RSA [26]. So we have $|A_{S,i}| = 160$ bits. Assume that each node ID is of l_{ID} bits, the hash seed x is of l_x bits, the packet overhead is as follows:

$$\begin{aligned}
\text{packet overhead} &= \frac{\alpha |\Upsilon_1|}{\text{len}} \\
&= \frac{\alpha \left(|A_{S,i}| + |\text{Mark}| + |\text{KN}_{1,z}| + |x|\right)}{\text{len}} \\
&= \frac{\alpha \left(160 + |\text{Mark}| + l_{\text{ID}} + l_x\right)}{\text{len}} \\
&= \frac{\alpha \left(160 + |\text{Mark}| + |\text{KN}_{1,z}| + |x|\right)}{|\Omega| + \alpha \left(160 + |\text{Mark}| + |\text{KN}_{1,z}| + |x|\right)} \\
&= \frac{1}{\varepsilon + 1},
\end{aligned} \tag{12}$$

where $\varepsilon = |\Omega|/(\alpha(160 + |\text{Mark}| + l_{\text{ID}} + l_x))$. In ELSP, $|\Omega|$ is fixed and unchangeable. Since $|\text{Mark}|$, l_{ID} and l_x are also constant, the packet overhead is in direct ratio to α, the number of packet layers, or path objects.

Now we discuss T, the number of end-to-end packet transmissions cost by each message from S to D. Suppose for the sake of simplicity that no two consecutive packet forwarders are the same, and that no key node is selected as a proxy node. Let L_i denote the number of times that packet Ψ_i (for all $i \in [1, \alpha]$) is transmitted before entering group i. According to Section 4.2.5, L_i satisfies the geometric distribution $\Pr(L_i = k) = (1 - \rho)\rho^{k-1}$, for all $k \geq 1$, with mean $\overline{L_i} = 1/(1 - \rho)$. Each Ψ_i also involves α intergroup transmissions, each for one proxy node in group i. According to the packet-forwarding process in Section 4.2.5, we can derive

$$T = \sum_{i=1}^{\alpha} \left(\overline{L_i} + 1\right) = \frac{\alpha}{1 - \rho} + \alpha. \tag{13}$$

Note that T includes $\alpha/(1 - \rho)$ intragroup transmissions and α intergroup transmissions. The former are hidden by intragroup traffic rate of λ packets/second and can be dynamically adjusted as needed. So, each message only incurs α inter-group transmissions. Besides, given a certain value of λ, the larger T it is, the fewer concurrent sessions can be supported and vice versa.

5.1.2. Computation Cost. In ELSP, the most time-consuming task is undoubtedly the pairing $\hat{e}$ operation for shared-key establishment, which chooses Tate paring [26]. Zhang et al. [29] quantify the energy consumption of the Tate pairing. It assumes that the sensor CPU is a low-power 32-bit Intel PXA255 processor at 400 MHz. The computation of the Tate pairing roughly needs $33/400 \times 752 \approx 62.04$ ms, and the energy consumption is approximately 25.5 mJ. In fact, the pairing function in the protocol is executed relatively rarely. Specifically, each pair of nodes only needs to do $\hat{e}$ once on demand to establish a shared key whereby to encrypt and authenticate intergroup or intragroup traffic using efficient symmetric key ciphers, the remaining pairing operations can be precomputed and stored for all protocol instances. Moreover, if source S intends to transmit multiple messages to destination D, each key node only need execute $\hat{e}$ once to compute a shared key when transmitting the first message. All subsequent packet processing are implemented based on the shared keys using efficient symmetric-key ciphers. Therefore, the computation cost of ELSP is totally acceptable even on low-end sensor device.

5.2. Security Analysis. Packets in ELSP use pseudonym instead of real source ID so that key nodes can not determine the initiators of received packets. Further, even the base station (a key node as well) cannot ascertain who sends it the message if the source node does not reveal its real identity in the message. This implies that the adversary can not directly determine packet sources. However, the adversary can assign a probability to each sensor node for being the source of a given packet. Therefore, we will investigate the resilience of ELSP against such probabilistic attacks.

We first use two entropy-based metrics to evaluate source anonymity. Then we describe the random forwarding approach of how to prevent internal attackers from determining key nodes. Finally, we evaluate the capability of ELSP providing source anonymity and show its efficiency compared with existing works.

5.2.1. Anonymity Metrics. Pfitzmann and Hansen [30] first defined anonymity as "the state of Indistinguishable within a set of all possible subjects, anonymity set." Since then, anonymity set has become a popular metric to evaluate

the anonymity in various anonymous communication systems. Generally, the greater the anonymity set is, the better anonymity achieved. However, it is unable to reflect the possibility that the adversary assigns different probabilities to each node as being the source of a given packet. This problem was solved by an entropy-based metric proposed in [31]. Briefly speaking, let Φ be a set of N nodes ($|\Phi| = N$) in an anonymous communication system; the anonymity entropy is defined as

$$\Delta = -\sum_{X=\Phi} P_X \log_2 (P_X). \tag{14}$$

Here, P_X represents the probability of the adversary assigned to node X being the source of a packet, and Δ denotes the uncertainty about which node is the source of a packet or the additional information that the adversary needs to determine the packet source. It follows that $0 \le \Delta \le \log_2(N)$ [32]. The upper bound is achieved when each node $X \in \Phi$ is assigned an equal probability of $1/N$ to be the source as viewed from the adversary (the ideal case); the lower bound is attained when S is assigned a probability of one, while each node $X \in \Phi \setminus \{S\}$ is assigned a probability of zero. In addition, Diaz et al. [33] defined anonymity degree as

$$\nu = \frac{\Delta}{\log_2 (N)}, \tag{15}$$

which indicates the distance between the real anonymity entropy and the maximum anonymity entropy that a system can provide.

5.2.2. Efficacy of Random Forwarding. We denoted source S a key node by $\mathrm{KN}_{0,z}$. It is crucial in ELSP to prevent the attackers from identifying key nodes $\mathrm{KN}_{i,z}$ ($0 \le i \le \alpha$). As discussed before, each packet Ψ_{i+1} takes a random path starting from key node $\mathrm{KN}_{i,z}$ before entering group $i+1$; this helps withstanding eavesdroppers (external attackers). However, there might be some sensor nodes compromised (internal attackers) in the transmission path. The countermeasure is to identify and isolate these compromised nodes via some effective solutions like reputation and trust-based mechanism as in [34] this can improve anonymous routing, but also increase the complexity of the system. In the following, we illustrate the efficacy of the random-forwarding mechanism against such internal attackers without other complements.

We assume that a set of $c \in [1, |g_i| - 1]$ compromised nodes in group g_i ($0 \le i \le \alpha - 1$), among which at least one appears in the forwarding path of packet Ψ_{i+1}. The goal of internal attackers is to find out which noncompromised node in g_i is $\mathrm{KN}_{i,z}$ that initiated Ψ_{i+1}. Let B denote the first compromised node which received Ψ_{i+1} from a noncompromised node A. From the point of view of internal attackers, all the noncompromised nodes in g_i other than A have equal probability to be $\mathrm{KN}_{i,z}$, but they are obviously less likely to be $\mathrm{KN}_{i,z}$ than A. We need to analyze how confident the adversary can be that A is indeed $\mathrm{KN}_{i,z}$, or in other words, the probability that they assigned to A being $\mathrm{KN}_{i,z}$.

Theorem 1. *Let one suppose that the first compromised node B in group i, for all $i \in [0, \alpha - 1]$, received packet Ψ_{i+1} from node A; the probability that the adversary assigned to A as the key node $KN_{i,z}$ is $(n - (n - c - 1)\rho)/n$, where $n = |g_i|$ and c is the number of compromised nodes in group i.*

Proof. Let $\overrightarrow{AB}$ be the event that B received packet Ψ_{i+1} from node A. The probability of A being $\mathrm{KN}_{i,z}$ is

$$\Pr\left(A = \mathrm{KN}_{i,z} \mid \overrightarrow{AB}\right) = \frac{\Pr\left(A = \mathrm{KN}_{i,z}, \overrightarrow{AB}\right)}{\Pr\left(\overrightarrow{AB}\right)}, \tag{16}$$

where

$$\begin{aligned}\Pr\left(\overrightarrow{AB}\right) &= \Pr\left(A = \mathrm{KN}_{i,z}, \overrightarrow{AB}\right) + \Pr\left(A \ne \mathrm{KN}_{i,z}, \overrightarrow{AB}\right) \\ &= \Pr\left(A = \mathrm{KN}_{i,z}\right) \cdot \Pr\left(\overrightarrow{AB} \mid A = \mathrm{KN}_{i,z}\right) \\ &\quad + \Pr\left(A \ne \mathrm{KN}_{i,z}\right) \cdot \Pr\left(\overrightarrow{AB} \mid A \ne \mathrm{KN}_{i,z}\right).\end{aligned} \tag{17}$$

Let $\mathrm{Pos}(B) = k$ be the event that B locates in the kth position of the path. Then $p'_k = \Pr(\overrightarrow{AB}, \mathrm{Pos}(B) = k \mid A = \mathrm{KN}_{i,z})$ represents the probability that A is indeed $\mathrm{KN}_{i,z}$, and Ψ_{i+1} went through $k - 1$ noncompromised nodes before A. There are three cases:

(i) $k = 1$: it shows that A selected B as the first packet forwarder; this occurs with probability $p'_k = 1/n$ because each packet forwarder is chosen randomly and uniformly from group $i(|g_i| = n)$.

(ii) $k = 2$: it shows that A selected itself as the first packet forwarder and then B as the second one, which occurs with probability $p'_2 = \rho/n^2$.

(iii) $k > 2$: it shows that packet Ψ_{i+1} passed $k - 2$ noncompromised nodes, and the last one selected A as the $(k - 1)$th packet forwarder who chose B as the kth packet forwarder, which occurs with $p'_k = \rho^{k-1} \cdot ((n-c)/n)^{k-2} \cdot (1/n^2) = (\rho/n^2)(((n-c)\rho)/n)^{k-2}$. It follows that

$$\begin{aligned}\Pr\left(\overrightarrow{AB} \mid A = \mathrm{KN}_{i,z}\right) &= \frac{1}{n} + \frac{\rho}{n^2} \sum_{j=0}^{\infty} \left(\frac{(n-c)\,\rho}{n}\right)^j \\ &= \frac{n - (n-c-1)\,\rho}{n^2 - n\,(n-c)\,\rho}.\end{aligned} \tag{18}$$

Likewise, we have

$$\begin{aligned}\Pr\left(\overrightarrow{AB} \mid A \ne \mathrm{KN}_{i,z}\right) &= \sum_{k=1}^{\infty} \Pr\left(\overrightarrow{AB}, \mathrm{Pos}\,(B) = k \mid A \ne \mathrm{KN}_{i,z}\right) \\ &= 0 + \sum_{k=2}^{\infty} \rho^{k-1} \cdot \left(\frac{n-c}{n}\right)^{k-2} \cdot \frac{1}{n^2} \\ &= \frac{\rho}{n^2 - n\,(n-c)\,\rho}.\end{aligned} \tag{19}$$

At last, if B has no other information, all the noncompromised nodes in group i are equally likely to be the key node $\mathrm{KN}_{i,z}$, so we have

$$\Pr\left(A = \mathrm{KN}_{i,z}\right) = \frac{1}{n-c}, \qquad \Pr\left(A \neq \mathrm{KN}_{i,z}\right) = \frac{n-c-1}{n-c}. \tag{20}$$

Substituting (18), (19), and (20) into (17) and then (16), we finally obtain

$$\Pr\left(A = \mathrm{KN}_{i,z} \mid \overrightarrow{AB}\right) = \frac{n-(n-c-1)\,\rho}{n}. \tag{21}$$

□

Based on (21), we can further deduce that

(i) when $n = c+1$, it indicates that all the nodes in g_i other than A are compromised, $\Pr(A = \mathrm{KN}_{i,z} \mid \overrightarrow{AB}) = 1$. Thus, the adversary can make sure that A is indeed $\mathrm{KN}_{i,z}$;

(ii) when $n > c+1$, the larger ρ, the smaller $\Pr(A = \mathrm{KN}_{i,z} \mid \overrightarrow{AB})$, the better the key node is concealed from the internal attackers, and the larger the communication overhead T.

Note that all the other $n - c - 1$ noncompromised sensor nodes have equal probability of being $\mathrm{KN}_{i,z}$. Consequently, the probability distribution of each node in g_i being $\mathrm{KN}_{i,z}$ as viewed by the adversary is given by

$$\Pr\left(X = \mathrm{KN}_{i,z}\right) = \begin{cases} \dfrac{n-(n-c-1)\,\rho}{n} & X = A, \\ \dfrac{\rho}{n} & X \in g_i,\ X \neq A, \\ 0 & \text{else.} \end{cases} \tag{22}$$

5.2.3. Source Anonymity Measurement. It should be noted that besides the event packets, there are still other communications among the sensor nodes involving a large number of packets, which makes it infeasible for the adversary to precisely separate different communication sessions from eavesdropping [1]. This allows us to focus on the impact of internal attackers. For simplicity, we still consider the session from source S to destination D.

We first discuss the impact of consecutively compromised key nodes. Suppose that the adversary compromised C out of the overall N sensor nodes. Because of the layered encryption, the same information appears entirely different across packet layers so that only key nodes $\mathrm{KN}_{i,z}$ ($0 \le i \le \alpha - 1$) that strip off one layer of encryption can correlate messages across two adjacent layers. Note that the last key node $\mathrm{KN}_{\alpha,z}$ does not further forward; the packet, therefore, is excluded here. We call a chain of e ($1 \le e \le \alpha - 1$) compromised key nodes participating in the same packet delivery as an e-chain. The e-chain can integrate the same message across $e + 1$ consecutive packet layers. For example, when $e = 1$, no consecutive key nodes were compromised, and when $e = \alpha-1$, the adversary can trace the packet from g_1 to g_α. It is possible that there exist multiple disjoint chains, but the adversary cannot link them together, so we only need to consider the longest e-chain, referred to the e^*-chain, which exposes the maximum information to the adversary. The e^*-chain can begin with any of the $\alpha - e^*$ key nodes, and the probability of it starting from $\mathrm{KN}_{i,z}$ ($1 \le i \le \alpha - 1$) is $1/(\alpha - e^*)$. In other words, there is at least one compromised node in g_{i-1} serving random packet forwarding; g_{i-1} is the source group g_0 with probability $1/(\alpha - e^*)$.

We also consider an extreme case where there is at least one compromised node serving random packet forwarding in each group. For convenience, we assume that each group comprises the same number of n sensor nodes (i.e., $N = Mn$) and the adversary knows α. There might be no e^*-chain; in this case, the anonymity of S will be definitely better than what we will demonstrate below. Theorem 2 gives the principle about the source anonymity.

Theorem 2. *Let the e^*-chain beginning with $\mathrm{KN}_{i,z}$ ($i \in [1, \alpha]$) and the first compromised node B in group g_{i-1} received packet Ψ_i from node A, the probability that the adversary assigned to A as the packet source is $(n-(n-c-1)\rho)/(n(\alpha-e^*))$, where c is the number of compromised nodes in g_{i-1}.*

Proof. The proof is directly based on Theorem 1. Since B is in source group g_0 with the probability of $1/(a - e)$, then its predecessor A is in the random-forwarding path being source S with the probability of

$$\begin{aligned} \Pr\left(A = S \mid \overrightarrow{AB}\right) &= \Pr\left(A = S \mid \overrightarrow{AB}, B \in g_0\right) \\ &= \frac{n-(n-c-1)\,\rho}{n} \cdot \frac{1}{(\alpha - e^*)} \\ &= \frac{n-(n-c-1)\,\rho}{(\alpha - e^*)\,n}. \end{aligned} \tag{23}$$

□

Likewise, each noncompromised node except A in g_{i-1} has the equal probability of $\rho/((\alpha-e^*)n)$ being the source. Let us suppose that there are C compromised nodes among all the N sensor nodes in the network (excluding source S and destination D) the residue $(M-1)n-(C-c)$ noncompromised nodes not in g_{i-1} equally hold the probability of $(\alpha-e^*-1)/(\alpha-e^*)$ being the source. In a word, the probability distribution of each node being the source from the adversary's point of view is given by

$$\Pr\left(X = S\right) = \begin{cases} \dfrac{n-(n-c-1)\,\rho}{(\alpha - e^*)\,n} & X = A, \\ \dfrac{\rho}{(\alpha - e^*)} & X \in g_{i-1}, X \neq A, \\ \dfrac{\alpha - \beta - 1}{(\alpha - e^*)\,(Mn - n - C + c)} & \text{else.} \end{cases} \tag{24}$$

Finally, we can obtain the source anonymity entropy using (14) and the source anonymity degree using (15).

5.3. Performance Evaluation. We first give numerical results to demonstrate the effectiveness of ELSP, and then make a comparison with existing schemes by simulations. Unless specified otherwise, we assume that each sensor node is compromised independently with probability 0.1, which is considered a severe situation. Due to space limitations, we omit the calculation details.

5.3.1. Numerical Results. Figure 2 illustrates the parameters α, e^*, and ρ impact on the source anonymity degree, where the network size $N = 400$, the number of groups $M = 20$, and $C = 40$ nodes are compromised. We can see that the larger α and ρ are, the higher the source anonymity is. In addition, given parameters α and ρ, increasing e^* will decrease the source anonymity. The reason is straight: the larger e^* is, the more key nodes are compromised by the adversary, which in turn, increases the probability of message disclosed. Note that with $\rho > 0$, it is impossible for the adversary to ascertain the packet source even when the first $\alpha - 1$ key nodes are all compromised. These results verify the efficiency of random-forwarding technique in improving source anonymity.

Figure 3 shows the impact of the total number of compromised nodes C on source anonymity, where $N = 400$, $M = 20$, $\alpha = 5$, $\rho = 0.7$. It is obvious that the more compromised the nodes are, the lower the source anonymity is; this coincides with the intuition. For example, when $C = 200$, namely, half of the sensor nodes are compromised; the source privacy are still higher than 0.9. Based on this figure, it can be concluded that ELSP is resilient to node compromise attack.

The source privacy is also relevant to the group size n (cf. (24)). Figure 4 shows the impact of the group size n on source anonymity, where $N = 1000$ and $C = 100$. As we can see, source anonymity increases as n increases; meanwhile, it increases the total intragroup traffic (communication overhead) and vice versa. In fact, various objects may have different security levels; in order to provide differentiated source anonymity requirement, one viable solution is to hide more important nodes with higher privacy requirements in relatively large groups while letting other groups be of small sizes. Therefore, ELSP has a nice property of flexibility of source anonymity that can be guaranteed by varying the group size n.

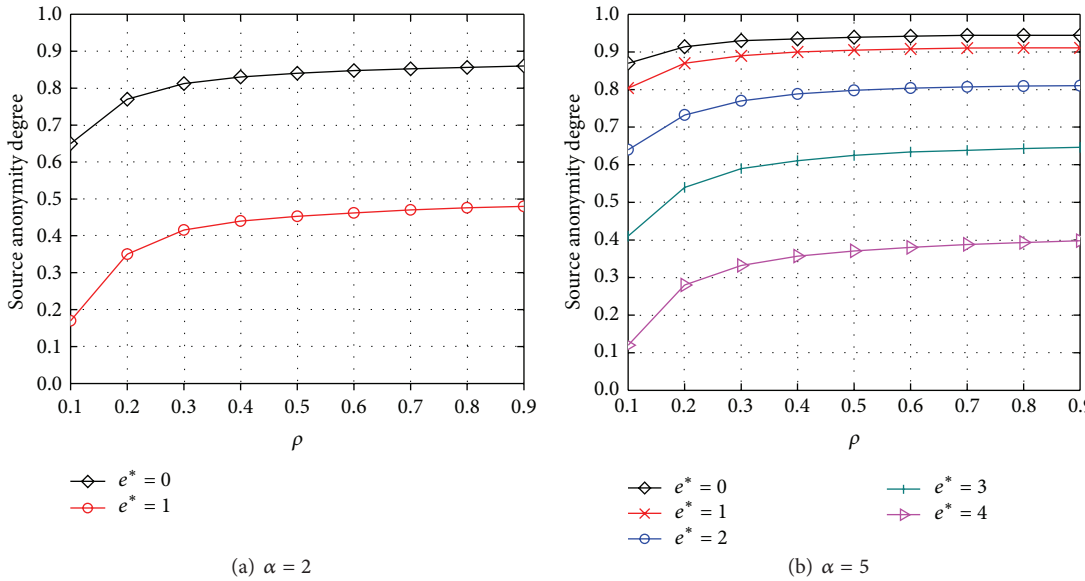
5.3.2. Comparison. In this subsection, we use both functional analysis and simulations to get a comprehensive comparison with DCARPS [16], random walk [5], and fake event packet generation [6, 8].

First, we compare the difference between DCARPS and ELSP from the technique implementation. In particular, DCARPS adopts label-switching approach, while ELSP uses the pseudonym and intermediate packet-altering scheme to conceal source traversal information. ELSP makes much less assumptions while considering a strong threat model as well as bearing the lightweight design in mind. DCARPS requires the base station to know the topology information of the network. ELSP only needs the information of intermediate nodes. In ELSP, only a small number of nodes are chosen to reconstruct the packet before forwarding, whereas in DCARPS, all the nodes involved in delivering the packet have to perform decryption and re-encryption operation with the new label. Further, in ELSP, this construction method allows the base station to verify the packet, while in DCARPS it is only used for routing. More importantly, node compromise attacks are not considered in DCARPS, which is the main attention we focus on ELSP.

Second, we conduct the simulation in NS2 to show the effectiveness of ELSP and compare it to random walk technique [5]. In Figure 5, we show the overhead (energy consumption) of random walk-based methods and ELSP. For fair comparison, sensor nodes in random walk-based methods are also organized into pairwise disjoint groups; the x-axis represents the longest path length of 50 hops between source and destination. Note that the overhead of random walk-based methods includes propagation over the random path and the journey towards the base station. The random walk lengths (denoted by H) are 10, 15, 20, and 25, respectively. The rabbling probability ρ is chosen as a stable status of 0.75 for ELSP, which is equivalent to $1/(1 - 0.5) = 4$ hops of intragroup random forwarding (cf. Section 5.1.1). As shown in Figure 5, the overhead of ELSP slightly increases with the path lengths, while the overhead involved in random walk-based methods is higher than ELSP.

In Figure 6, we set $N = 1000, C = 100, n = 10$ for random walk-based methods and ELSP. We also define privacy in random walk based methods as the length of distance the random walk takes the source information away from the actual source when compared to the path length (say S-D distance). The source privacy achieved for ELSP and random walk based scheme with the same path lengths and random walk lengths are shown in Figure 5. As we can see, the source anonymity degree or privacy of ELSP is constant irrespective of the path lengths, while the random walk based methods' privacy decreases with the increases of path length. This is because we consider a rather strong threat model, where the adversary can either eavesdrop over a larger coverage area or compromise sensor nodes. Thus, once the attackers obtain data packets on compromised node, they can traceback to the source with higher probability. Conversely, in ELSP, each packet is modified en route by selected nodes to make it difficult for the adversary to trace back to the source.

Finally, Figure 7 shows the overhead incurred in fake packet generation schemes [6, 8] compared to ELSP. We consider different numbers of fake packet-generating nodes (denoted by F), with the numbers ranging from 20% to 50% of all nodes. The x-axis represents the number of actual source nodes in the network as a percentage of all nodes. As shown in Figure 7, the larger the source nodes in the network are, the smaller the overhead incurred. We can conclude that any form of fake packet generation technique will have significantly higher overhead compared to ELSP. Even the conservative case of 20% of sensor nodes generating fake packets leads to higher overhead compared to ELSP. This is because the fake packet generation is an obfuscating technique which is only successful under the heavy load



(a) $\alpha = 2$ (b) $\alpha = 5$

FIGURE 2: Impact of α, ρ, and e^* on source anonymity.

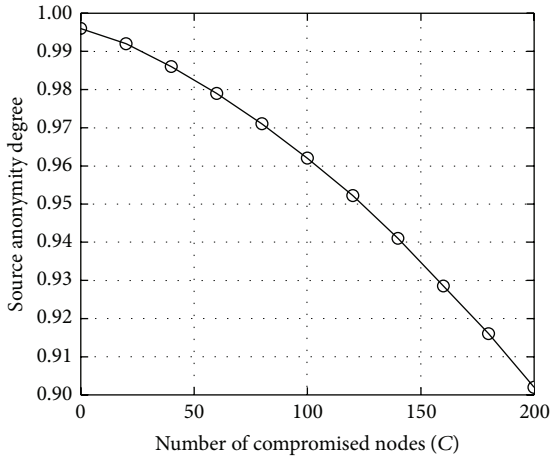


FIGURE 3: Impact of C on source anonymity.

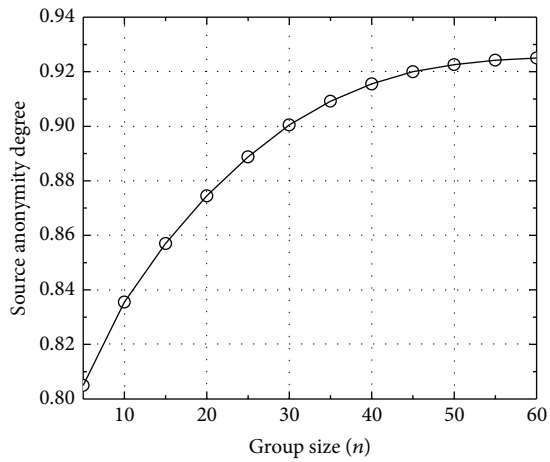


FIGURE 4: Impact of n on source anonymity.

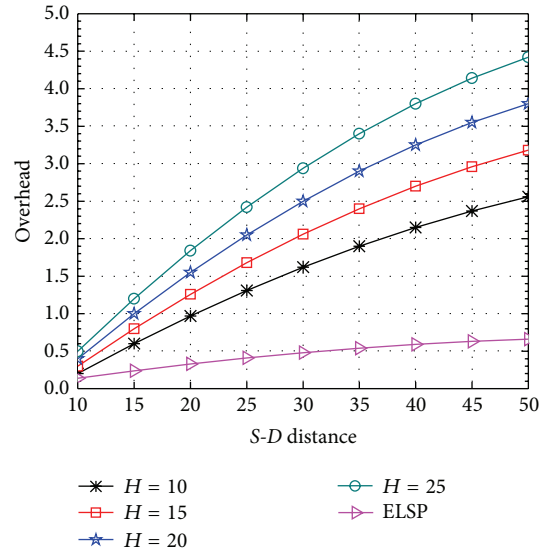


FIGURE 5: Overhead comparison between random walk-based schemes and ELSP.

of fake packet generation. When a sensor node detects an event and generates the event reporting packet, there should be at least one more source node generating a fake event-reporting packet. On this occasion, we see that 50% of the traffic corresponds to fake traffic. Moreover, to increase security will incur more overhead in the form of fake packets. Consequently, the realization of such a system is dependent on the amount of fake packets generated which still provide very minimal source privacy and have more overhead than ELSP. In addition, the fake packet generation schemes do not consider node compromises.

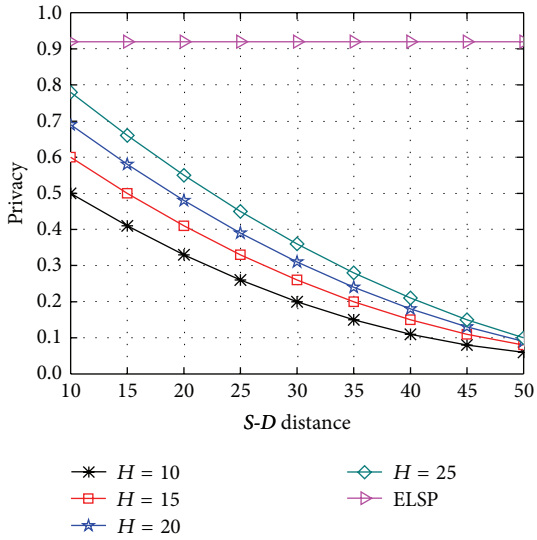


Figure 6: Privacy comparison between random walk-based schemes and ELSP.

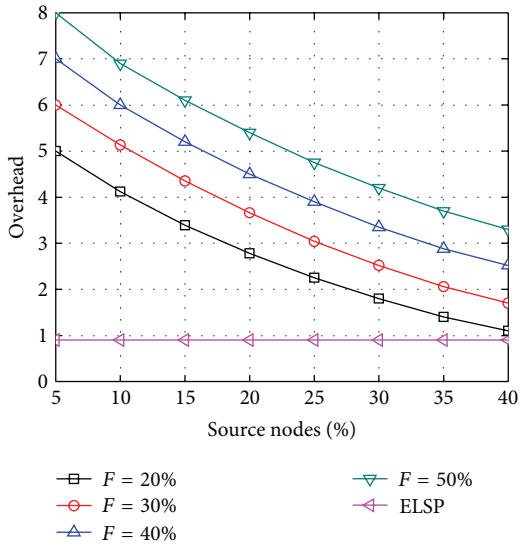


Figure 7: Overhead comparison between fake packet generation schemes and ELSP.

6. Conclusions

Sensor node detecting the event is important, as it can reveal the event occurrence location and time occurrence; thus, needs to maintain source privacy. Previous works consider an eavesdropping adversary and do not provide countermeasures to an intrusive node compromise attack with global eavesdropping capabilities. In this paper, we presented the design and evaluation of a novel anonymous mechanism for WSNs. By utilizing the grouping, the self-generated pseudonym, and the Identity-Based Cryptography, the proposed protocol is demonstrated to achieve desired security objectives and efficiency. As future work, we will seek to analyze the security of our scheme under other adversary models.

Acknowledgments

The authors acknowledge support from the National Natural Science Foundation of China (Grant nos. 61173136, 61202462, 61173141, and 61232016), and the fund support of the key subject of Fujian province—Computer Application Technology.

References

[1] J.-F. Raymond, "Traffic analysis: protocols, attacks, design issues, and open problemsin," in *Proceedings of the International Workshop on Design Issues in Anonymity and Unobservability*, pp. 10–29, Berkeley, Calif, USA, 2000.

[2] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–88, 1981.

[3] A. Pfitzmann and M. Waidner, "Networks without user observability," *Computers and Security*, vol. 6, no. 2, pp. 158–166, 1987.

[4] M. Reiter and A. Rubin, "Crowds: anonymity for web transactions," *ACM Transactions on Information and System Security (ITSSEC)*, vol. 1, no. 1, pp. 66–92, 1998.

[5] 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*, pp. 599–608, Columbus, Ohio, USA, June 2005.

[6] 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.

[7] 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)*, pp. 1–10, Istanbul, Turkey, September 2008.

[8] M. Shao, Y. Yang, S. Zhu, and G. Cao, "Towards statistically strong source anonymity for sensor networks," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 466–474, Phoenix, Ariz, USA, April 2008.

[9] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, 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 (WiSec '08)*, pp. 77–88, Alexandria, Va, USA, April 2008.

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

[11] N. Li, M. Raj, D. Liu, M. Wright, and S. K. Das, "Using data mules to preserve source location privacy in wireless sensor networks," in *Proceedings of the 13th International Conference on Distributed Computing and Networking (ICDCN '12)*, vol. 7129 of *Lecture Notes in Computer Science*, pp. 309–324, Hong Kong, Hong Kong, 2012.

[12] Y. Li, J. Li, J. Ren, and J. Wu, "Providing hop-by-hop authentication and source privacy in wireless sensor networks," in *IEEE Conference on Computer Communications (INFOCOM '12)*, pp. 3071–3075, Orlando, Fla, USA, 2012.

[13] W. Zhang, N. Subramanian, and G. Wang, "Lightweight and compromise-resilient message authentication in sensor networks," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 2092–2100, Phoenix, Ariz, USA, April 2008.

[14] M. Albrecht, C. Gentry, S. Halevi, and J. Katz, "Attacking cryptographic schemes based on perturbation polynomials," Cryptology ePrint Archive, Report 2009/098, 2009, http://eprint.iacr.org/.

[15] 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.

[16] A. A. Nezhad, A. Miri, and D. Makrakis, "Location privacy and anonymity preserving routing for wireless sensor networks," *Computer Networks*, vol. 52, no. 18, pp. 3433–3452, 2008.

[17] G. Chai, M. Xu, W. Xu, and Z. Lin, "Enhancing sink-location privacy in wireless sensor networks through k-anonymity," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 648058, 16 pages, 2012.

[18] C. Y. Chow, M. F. Mokbel, and T. He, "A privacy-preserving location monitoring system for wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 94–107, 2011.

[19] R. Di Pietro and A. Viejo, "Location privacy and resilience in wireless sensor networks querying," *Computer Communications*, vol. 34, no. 3, pp. 515–523, 2011.

[20] C. Li and M. Hwang, "A lightweight anonymous routing protocol without public key en/decryptions for wireless Ad Hoc networks," *Information Sciences*, vol. 181, no. 23, pp. 5333–5347, 2011.

[21] N. T. T. Huyen, M. Jo, T.-D. Nguyen, and E.-N. Huh, "A beneficial analysis of deployment knowledge for key distribution in wireless sensor networks," *Security and Communication Networks*, vol. 5, no. 5, pp. 485–495, 2012.

[22] B. Zhou, S. Li, Q. Li, X. Sun, and X. Wang, "An efficient and scalable pairwise key pre-distribution scheme for sensor networks using deployment knowledge," *Computer Communications*, vol. 32, no. 1, pp. 124–133, 2009.

[23] D. Du, H. Xiong, and H. Wang, "An efficient key management scheme for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 406254, 14 pages, 2012.

[24] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[25] A. K. Das, "ECPKS: an improved location-aware key management scheme in static sensor networks," *International Journal of Network Security*, vol. 7, no. 3, pp. 358–369, 2008.

[26] P. Barreto, H. Kim, B. Bynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," in *Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '02)*, pp. 354–368, Santa Barbara, Calif, USA, 2002.

[27] M. K. Wright, M. Adler, B. N. Levine, and C. Shields, "The predecessor attack: an analysis of a threat to anonymous communications systems," *ACM Transactions on Information and System Security*, vol. 7, no. 4, pp. 489–522, 2004.

[28] G. Danezis, C. Diaz, E. Kasper, and C. Troncoso, "The wisdom of crowds: attacks and optimal constructions," in *Proceedings of 14th European Symposium on Research in Computer Security (ESORICS '09)*, pp. 406–423, Saint-Malo, France, 2009.

[29] 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.

[30] A. Pfitzmann and M. Hansen, "Anonymity, unobservability, and pseudonymity: a consolidated proposal for terminology," Draft v0.25, 2005.

[31] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Proceedings of the 2nd International Conference on Privacy Enhancing Technologies (PET '02)*, pp. 41–53, Heidelberg, Germany, 2002.

[32] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley, London, UK, 2nd edition, 2006.

[33] C. Diaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Proceedings of the 2nd International Conference on Privacy Enhancing Technologies (PET '02)*, pp. 54–68, Heidelberg, Germany.

[34] G. V. Crosby, L. Hester, and N. Pissinou, "Location-aware trust-based detection and isolation of compromised nodes in wireless sensor networks," *International Journal of Network Security*, vol. 12, no. 2, pp. 107–117, 2011.