Source-Location Privacy Protection Based on Anonymity Cloud in Wireless Sensor Networks

Na Wang, Junsong Fu, Jian Li, and Bharat K. Bhargava, *Fellow, IEEE*

***Abstract*— An adversary can deploy parasitic sensor nodes into wireless sensor networks to collect radio traffic distributions and trace back messages to their source nodes. Then, he can locate the monitored targets around the source nodes with a high probability. In this paper, a Source-location privacy Protection scheme based on Anonymity Cloud (SPAC) is proposed. We first design a light-weight (t, n)-threshold message sharing scheme and map the original message to a set of message shares which are shorter in length and can be processed and delivered with minimal energy consumption. Based on the shares, the source node constructs an anonymity cloud with an irregular shape around itself to protect its location privacy. Specifically, an anonymity cloud is a set of active nodes with similar radio actions and they are statistically indistinguishable from each other. The size of the cloud is controlled by the preset number of hops that the shares can walk in the cloud. At the border of the cloud, the fake source nodes independently send the shares to the sink node through proper routing algorithms. At last, the original message can be recovered by the sink node once at least t shares are received. The simulation results demonstrate that the SPAC can strongly protect the source-location privacy in an efficient manner. Moreover, the message sharing mechanism of SPAC increases the confidentiality of network data and it also brings high tolerance for the failures of sensor nodes to the data transmission process.**

***Index Terms*— Source-location privacy protection, data confidentiality, anonymity cloud, message sharing, wireless sensor networks.**

I. Introduction

Wireless sensor networks (WSNs) are composed of a large number of smart devices that collaborate with each other to perform various tasks. Due to the developments in sensor technology, circuit engineering, and information techniques, WSNs have been widely used in many fields, including wild habitat monitoring, target tracing and military surveillance [1]–[3]. In general, once the information is collected, it will be immediately delivered to the sink node in a multi-hop manner and then the information can be used by the network operator. The networks are likely to be deployed in harsh environments and all the nodes are strictly limited in resources such as energy, communication, computing, and storage capabilities. Meanwhile, some nodes may not function properly and fail to monitor the environment or to receive and transmit packets. It is a great challenge to design data collection schemes for WSNs which need to be light-weight, reliable, and robust.

WSNs are vulnerable to many threats [4]–[6]. Though numerous encryption and decryption techniques have been used to protect the security of data and networks [7]–[10], some contextual-information-based attacks cannot be processed properly. As a novel back-tracing attack, Hotspot-Locating attack proposed in [11] is a huge threat to source-location privacy. In WSNs, a source-location is defined as the location of the node that keeps the target monitored [12] and source-location privacy is the confidentiality of the source node's location. Moreover, a set of neighboring source nodes form a hotspot that generates a large data transmission amount causing an obvious inconsistency in the network traffic. Once a hotspot is located, a set of source nodes can be found. In general, Hotspots can be formed for different reasons, e.g., when the monitored wild animals have high density or spend some time in one area due to the availability of food, water, shadow, shelter, etc. The nodes in WSNs are wirelessly linked and hence the adversary can detect the radio distribution through a spectrum analyzer. Considering that a sensor node keeps silent for most of time until targets are detected, the adversary can easily trace back to the source nodes by analyzing the radio behaviors of the nodes in the networks. At last, he can locate the surrounding targets monitored by the source nodes with a high probability. It can be observed that Hotspot-Locating attack is easy to implement with a low cost and it is a huge threat to WSNs.

As an example, a wildlife protection organization deploys a WSN to monitor wild pandas [13] and the collected information is periodically reported to the sink node for further analysis. In this scenario, the hunters can locate the pandas through Hotspot-Locating attack and apparently, this is a great threat to the pandas. Consequently, it is very meaningful to design source-location privacy protection schemes. For convenience,

Manuscript received August 6, 2018; revised November 25, 2018, March 27, 2019, and May 20, 2019; accepted May 23, 2019. Date of publication May 27, 2019; date of current version September 11, 2019. This work was supported in part by the Fundamental Research Funds for the Central Universities and in part by the China Postdoctoral Science Foundation under Grant 2019M650020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Aris Gkoulalas Divanis. *(Corresponding author: Junsong Fu.)*

N. Wang and J. Li are with the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: nawang@bupt.edu.cn; lijian@bupt.edu.cn).

J. Fu is with the National Engineering Laboratory for Mobile Network Technologies, School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: fujs@bupt.edu.cn).

B. K. Bhargava is with the Department of Computer Science, Purdue University, West Lafayette, IN 47906 USA (e-mail: bbshail@purdue.edu).

Digital Object Identifier 10.1109/TIFS.2019.2919388

we use pandas to represent the monitored target in the rest of this paper, though the target can be any monitored object in real WSNs.

In source-location privacy protection schemes, two adversary models, i.e., global adversary and local adversary, are widely employed [11], [14]–[18]. Global adversaries are assumed to be capable of monitoring the whole network and know all the radio transmissions in the data link layer. This model is impractical for extremely large WSNs. Moreover, if the adversaries can monitor the whole WSN, i.e., deploying a parasitic sensor network (PSN) with a similar size to that of the WSN, they can directly locate the targets (e.g., the pandas) by the PSN. How to stop the adversaries from locating targets directly by a PSN is very challenging and however this is not what we mainly concern in this paper. As a consequence, we employ the local adversary model which assumes that the adversaries have limited overhearing capability and a parasitic node can only monitor the local area at a given time. In general, the overhearing range R_o of the parasitic nodes is similar to the communication range of the sensor nodes R_c and for convenience, we set R_o equals to R_c in this paper.

In the common back-tracing attack, once a parasitic node monitors a package transmission made from node A, it moves to A and waits until another package is sent from node B. Then, it moves to B and waits to find another package transmission. The parasitic node repeats the above process until it locates the source node. Random routing algorithms [17], [18] can be employed to defend this attack. However, Hotspot-Locating attack is much stronger and the adversary uses traffic inconsistency caused by hotspot areas to locate pandas by analyzing the data collected by parasitic nodes. Though random routing algorithms can change the routing paths, they cannot hide the traffic inconsistency between hotspot areas and normal areas. Consequently, it is severe to propose novel source-location privacy protection approaches.

In this paper, we enhance Hotspot-Locating attack model to make it more practical and stronger. Then, to defend against the enhanced Hotspot-Locating attack, this paper proposes a source-location privacy protection scheme based on anonymity cloud named SPAC. We first design a lightweight (t, n)-threshold message splitting and sharing scheme particularly for WSNs based on congruence equations [19]. When a message $M$is generated by the source node, we map M to a set of independent sub-messages $s_1, s_2, \ldots, s_n$ called shares of M which are much shorter in length. Compared with message M, the shares can be processed and transmitted in the anonymity cloud flexibly with much less energy consumption. Each share contains part information of M and any subset with at least $t(1 \leq t \leq n)$ shares can reconstruct M, or otherwise M is safe in confidentiality.

Based on the shares, we construct an anonymity cloud around the source node to hide its accurate location. In SPAC, an anonymity cloud is a set of active nodes with similar radio behaviors and the nodes in a cloud are statistically indistinguishable. Note that, the size of a cloud is greatly larger than that of a hotspot and much smaller than that of the whole network. Given n shares, the source node randomly selects a hop number h_i for each share s_i which indicates how many steps the share can walk in the cloud. Then the source node sends the shares along with the hop numbers to its neighbors randomly. In the process of expanding the cloud, the next hop of a share is chosen through sector-based directed random walk model rather than unbiased random walk model to avoid conflicts between walking steps. Meanwhile, some fake shares are also generated and transmitted along with the real shares to protect radio traffic privacy. Once an anonymity cloud is constructed by a source node, all the other nodes in the cloud need not to construct a new cloud and they just need to follow the rules about radio transmission behaviors.

To destroy the transmission patterns underneath these shares, a random time delay is generated for each share at each transmitting step. We generate time delays from a normal distribution for the fake shares. Moreover, we set the time delays for the real shares by the method proposed in [14] to increase the freshness of the real shares while guaranteeing their security. Once a sensor node receives a real share that has been delivered for h hops, the node is defined as a fake source node and it delivers the share to the sink node through proper routing algorithms immediately. All the fake source nodes of message M locate near to the boundary of the cloud and they are naturally dispersive. This increases the difficulty for the adversaries to trace back. Once at least t shares are received, the sink node can reconstruct M and the message transmitting process is completed.

In this paper, it is an interesting attempt to introduce the message sharing technique into source-location privacy protection schemes. This brings mainly three important advantages to SPAC. First, message sharing scheme improves the confidentiality of data delivered in the network and it protects the source-location privacy indirectly. In SPAC, the adversary cannot recover M even some shares are captured and this increases the difficulty of extracting the source-location information from M. Second, without shares, the nodes in the cloud need to transmit the original message M and fake messages with the same length of M. Apparently, it generates an extremely large data transmission amount, especially for a huge cloud. In our scheme, we replace the original messages by real shares and fake shares that are much shorter in length. In this case, energy-efficiency of the whole networks greatly improves which makes SPAC more practical. Third, message sharing scheme significantly improves the robustness of the package delivery process. This can be explained by the fact that the sink node can recover message M if it receives just t shares though some other shares are lost.

Simulation results show that SPAC can provide strong protection on source-location privacy and it is much more energy-efficient compared with existing cloud-based schemes and global-adversary-based schemes. In addition, the proposed scheme is of high fault tolerance on the failure of sensor nodes and hence it can provide a reliable data transmission process with proper t and n.

The main contributions of this paper are summarized as follows:

- We present a novel network structure to show the back-tracing threat model clearly and then enhance the Hotspot-Locating attack model to make it stronger and more practical.
- We design a light-weight message sharing scheme particularly for WSNs based on congruence equations and maps the original message to a set of shorter shares which can be processed and delivered efficiently.
- An anonymity cloud with an irregular shape is constructed based on the shares to hide the accurate location of the real source node. The radio actions of the nodes are carefully designed to make them indistinguishable.
- A series of simulations are conducted to compare SPAC with existing routing-based schemes and cloud-based schemes in terms of source-location privacy protection, energy efficiency, and reliability.

The rest of this paper is organized as follows: A thorough review of the source-location privacy protection approaches and message sharing schemes is presented in Section II. Network model and the enhanced Hotspot-Locating attack model are presented in Section III. SPAC is presented in Section IV. The performance of SPAC is evaluated in Section V. We conclude this paper and mention our future research plan in Section VI.

II. Related Work

A. Source-Location Privacy Protection Schemes

In global-adversary-based schemes, the adversaries can analyze the source locations based on all the traffic information of the entire network [14], [15], [20]. In this case, the best choice to defend against the back-tracing attack is sending dummy messages to confuse the adversaries. Most existing approaches attempt to find good balances among the security of source node, the overhead of dummy messages and message delivery delay. Alomair *et al.* [14] proposed the notion of "interval indistinguishability" in their scheme and mapped the source node anonymity problem to the statistical problem of binary hypothesis testing to minimize time delay while ensuring the security of the source node. To reduce messages delay time without an apparent increase of data transmission amount, Shao *et al.* [15], [20] proposed a statistically strong source location privacy protection scheme. In this scheme, the source node sends real messages as soon as possible while keeping them indistinguishable from the dummy messages. To decrease the overhead of dummy messages, Lu et al. [21] designed a scheme for cluster-based WSNs in which the cluster heads collect information from its members periodically and filter out the dummy messages. Then, only the real messages are sent to the sink node. However, it results in a longer time delay because of the fixed packages sending rate. In [16], the dummy messages are filtered out by proxy nodes rather than cluster heads to further decrease message transmission amount. For different proxy assignment methods, the lifetimes of the networks are thoroughly analyzed. Mehta *et al.* [22] computed a lower bound on the communication amount needed for a given level of location privacy and then provided two techniques to balance time delay and data transmission amount.

Most local-adversary-based source location protection schemes use random routing algorithms to make the routing paths more difficult to be traced backward. The phantom routing technique [17] is a classic random routing technique and it is composed of two phases: a random walk phase and a subsequent single path routing phase. Initially, a package is sent out by the source node and it randomly walks for k steps which are preset by the operators. To avoid the random walk steps canceling each other, some directed random walk models are designed. Then, the random walk phase is transformed into the single path routing phase. The fake source node can employ any existing routing algorithm to deliver the messages to the sink node. Wang and Zeng [18] proposed a random routing algorithm called ARR, in which the source node can predefine the rough shape of the routing path by randomly selecting a set of virtual locations which can decide a set of agent nodes on the path. Then, an extremely complicated mechanism is designed to guarantee that the packages can be properly sent to the sink node by the agent nodes in a relay manner. Except for routing-based schemes, some other schemes are also proposed. Recently, a cloud-based scheme is proposed [11] and it is strongly related to SPAC we propose in this paper. In [11], the cloud is filled with fake messages and it is constructed through a complex process. Though the adversaries can trace back to the boundary of the cloud, it is difficult to find the accurate location of the source node. However, this scheme is very energy-consuming and its performance can be further improved.

SPAC also improves the confidentiality of data transmitted in the network and some related work in this field are discussed as follows. Liu et al. [10] proposed a secure and energy-efficient multi-path routing algorithm for the message shares in WSNs. This algorithm first randomly delivers the shares all over the network before sending them to the sink node. Simulation results show that it performs well even in the network with black holes. Ozdemir et al. [7] integrate false data detection with data aggregation to improve data confidentiality. Specifically, the sensor nodes between two consecutive data aggregators verify the data integrity on the encrypted data rather than the plaintext data to support confidential data transmission. Mahmoud *et al.* [8] designed a secure and reliable routing protocol for WSNs by combing payment and trust systems. The protocol always chooses those highly-trusted nodes having sufficient energy to minimize the probability of breaking the route. Ren *et al.* [23] attempt to design a location-aware end-to-end security data transmission framework for large-scale static WSNs. In this scheme, secret keys of the nodes are bound to geographic locations and each node stores a few keys. This location-aware property effectively limits the impact of compromised nodes only to their vicinity without affecting end-to-end data security.

B. Message Sharing Schemes

A message sharing scheme allows one to map an original message M to several independent messages $s_1, s_2, \cdots, s_n$, called shares. The shares can be distributed to a set of users P and only certain qualified subsets of users can recover M. The collection of all the qualified sets of users is defined as the access structure Γ. Since Shamir [24] first proposed the message sharing scheme with a threshold access structure, many researchers have contributed to message sharing methods and implementations. A comprehensive introduction of message sharing schemes is provided in [25], [26]. A (t, n)-threshold secret sharing scheme is a method of sharing a message M among a set of n users, in such a way that any t users can reconstruct the message M, but no group which contains less than t users can do so. Many classical (t, n)-threshold schemes have been proposed [27]–[30]. Marek and Urszula [31] presented a new message sharing algorithm based on the use of mathematical methods. This algorithm can be used as a new method or an intelligent component for message sharing. Zhang and Zhang [32] investigated the verifiable secret sharing scheme and they present an information-theoretical secure VSS scheme which can be further improved in terms of efficiency. A linear threshold verifiable secret sharing in bilinear groups is proposed in [26] and it is simple and energy-efficient. Huang *et al.* [33] computed a tight lower bound on the amount of communication amount between the users and the parties. Further, they generalized Shamir's secret sharing scheme and proposed a simple and efficient scheme with minimal communication. Harn and Fuyou [34] proposed a multilevel threshold secret sharing scheme based on the Chinese Remainder Theorem. Some secret sharing schemes have been employed in distributed sensor networks to enhance data confidentiality [6]. However, these schemes cannot be directly employed to protect the source-location privacy and they are too energy-consuming. This can be explained by the fact that most message sharing schemes are of high computation complexities and the lengths of the shares are too large.

III. Network Model and Enhanced Hotspot-Locating Attack

A. Network Model

In this paper, we consider a huge 2-D network composed of a large number of homogeneous sensor nodes. Each node in the network is assumed to be able to locate itself in proper manners [12], [35]. They can further get their neighbors' locations easily based on simple beacon communications. We further assume that each node is capable of computing, communication, and storage to properly execute the instructions. In general, a sink node acts as a bridge between a network and the network operator and it is much more powerful than the common nodes [36]. Therefore, we assume that the sink nodes in our network have sufficient resources in terms of computing, storage, and data transmission.

The deployed nodes in the network employ the k-Nearest neighbors tracing approach [37] to monitor the targets. Specifically, each node follows a sleeping schedule and keeps silent when no target is detected. However, if a node detects a target in its region of responsibility, it needs to keep active until the target moves out of the region. In general, a target is simultaneously detected by a set of nodes and we assume that these nodes can locate the target accurately in a cooperative manner. At last, the information of the target is sent to the sink nodes in time.

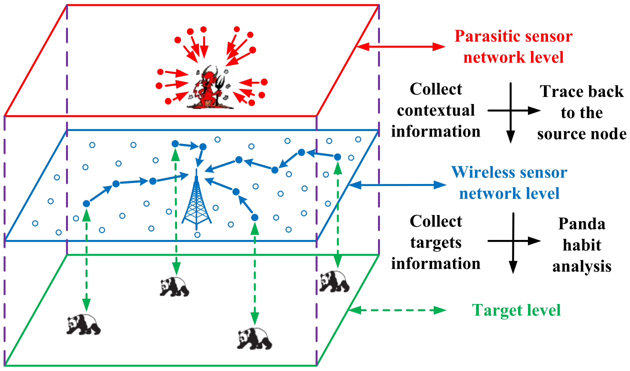
Parasitic sensor network level
Collect contextual information
Trace back to the source node
Wireless sensor network level
Collect targets information
Panda habit analysis
Target level

Fig. 1. Three levels of the whole network.

Similar to [17], [27], we consider a scenario that a hunter attempt to trace back to the source nodes and find the panda. To present the back-tracing threat clearly, we decompose the whole network into three levels, i.e., target level, WSN level, and PSN level, as shown in Fig. 1. The target level is the foundation of the whole network. Individuals or organizations need to deploy WSNs to collect the information about the targets such as the locations and physical conditions which can be used by the zoologist to analyze the habit of the pandas. To find the panda, the adversary deploys some parasitic nodes into WSNs to locate the source nodes. Once the source nodes are located, the pandas can be found with a high probability considering that the source nodes are naturally near to the pandas. All the parasitic nodes compose the PSN level. Compared with searching the pandas randomly in a boundless wild area, back-tracing attack based on PSN greatly improves the efficiency of the searching process.

B. Enhanced Hotspot-Locating Attack

We assume that the parasitic nodes are well equipped with modules of power, movement, communication, spectrum sensing and analysis, storage and computing. Each parasitic node can monitor radio signals locally and locate the sender of the messages. However, they cannot locate the receiver of the packages, because any node in the transmission range can be the receiver. The parasitic nodes can communicate with each other by wireless links and they can share the collected data in time. In this way, a set of parasitic nodes in a near area can form a more powerful organization and the monitor radius greatly enlarged compared with a single node.

In this paper, we employ an enhanced attack model of Hotspot-Locating attack model proposed in [11]. As an example, the process of Hotspot-Locating attack is presented in Fig. 2. A parasitic node is initially deployed around the sink node and some others are distributed in the network randomly. In back-tracing phase, the parasitic nodes collect traffic information including the coordinates of the nodes that sent a packet and the time of sending the packet. Then the parasitic nodes analyze the collected information and judges whether they find a hotspot or they can move to a more promising area that can lead to the hotspot. Two types of information including time correlation and packet sending rate are analyzed simultaneously to locate the hotspot.

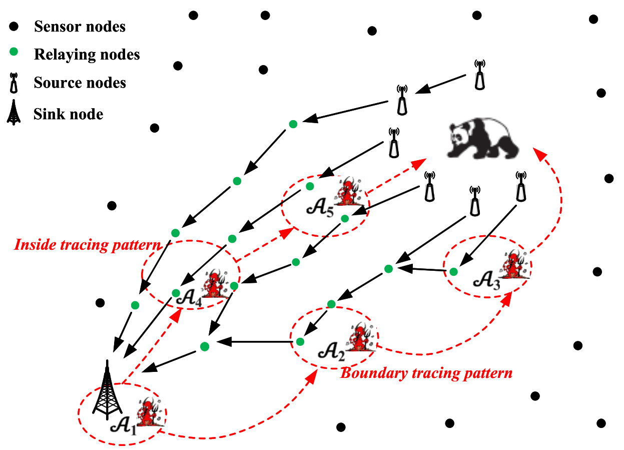

Fig. 2. Hotspot-locating attack.

Specifically, the adversary identifies a hotspot by using the fact that more packets are sent out by the nodes near to the hotspot compared with the nodes far away from the hotspot. Therefore, the adversary can continuously move toward the hotspot by analyzing the traffic rather than track back by a packet. As shown in Fig. 2, Hotspot-Locating attack comprises of two patterns including inside back-tracing pattern and boundary back-tracing pattern. In the inside back-tracing pattern, the parasitic nodes follow the high packet sending rates of the nodes which relay the hotspot's packets and finally reach a suspect region. In Fig. 2, a parasitic node moves from area $\mathcal{A}_4$ to $\mathcal{A}_5$ by employing the inside back-tracing pattern. Apparently, if a parasitic node moves out of the hotspot, the packet sending rate greatly decreases suddenly and hence it can infer the hotspot region. In the boundary back-tracing pattern, the parasitic nodes can identify the boundary easily by observing the large difference in packet sending rates between the two sides of the boundary. The parasitic nodes move on the boundary of a large packet sending rate until they reach a suspected region. In Fig. 2, a parasitic node moves from area $\mathcal{A}_1$ to $\mathcal{A}_2$ and then to $\mathcal{A}_3$ by employing the boundary back-tracing pattern.

Once the adversary finds a small suspect area, we assume that the adversary collects all his resources and deploy them in this area. In the extreme case, the adversary can monitor all the nodes in suspect. Consequently, we can treat the adversary as a global adversary and he knows all data transmission behaviors in this area. This assumption makes the Hotspot-Locating attack proposed in [11] more practical and stronger and it increases the difficulty of source-location privacy protection. We call this model the enhanced Hotspot-Locating attack model.

It can be observed that the proposed attack model in this paper is much stronger than packet-based back-tracing attack. Existing global-adversary-based source-location privacy protection schemes consume an extremely large data transmission amount and they are impractical for large WSNs. Though random routing algorithms are energy-efficient, they cannot defend against this attack effectively. As a consequence, it is of great importance to design an efficient and effective scheme.

IV. Source-Location Privacy Protection Based on Anonymity Cloud

In this section, we present the pre-deployment phase of WSNs in Section IV.A. Then, a light-weight message sharing scheme is particularly designed for WSNs based on congruence equations, and two important properties of this scheme are stated through provable Theorems in Section IV.B. The method of constructing the anonymity cloud based on the shares is presented in Section IV.C. Finally, we discuss the method of delivering the shares from the fake source nodes to the sink nodes and how to reconstruct the original messages based on the received shares at the sink nodes in Section IV.D.

A. Pre-Deployment Phase

To protect data privacy between a pair of nodes, we first design a pairwise key negotiation algorithm based on bilinear map. Before scattering all the sensor nodes into the monitored area, each node n_i is assigned with a unique identifier ID_{n_i}, a public key PK_{n_i} and a secret key SK_{n_i} which are used to negotiate session keys with its neighbors. Let $\mathbb{G}_0$ and $\mathbb{G}_1$ be two multiplicative cyclic groups of prime order p. Let g be a generator of $\mathbb{G}_0$ and e be a bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$ with the following properties:

1. Bilinearity: $\forall u, v \in \mathbb{G}_0$ and $a, b \in \mathbb{Z}_p$, $e\left(u^a, v^b\right) = e(u, v)^{ab}$.
2. Non-degeneracy: $e\,(g, g) \neq 1$.

Let H be a hash function and H :$\{0, 1\}^* \to \mathbb{G}_0$. The public key of node n_i is calculated as follows:

$$PK_{n_i} = \mathrm{H}(ID_{n_i}) \tag{1}$$

Private key generator (PKG) randomly chooses a master key s from $\mathbb{Z}_p^*$. The secret key of node n_i is calculated by PKG as follows:

$$SK_{n_i} = PK_{n_i}^s \tag{2}$$

Note that, though node knows PK_{n_i} and SK_{n_i}, it cannot obtain s because of discrete logarithm difficulty. Similarly, node n_j has the public key PK_{n_j} and the secret key SK_{n_j}.

In the deployed network, a pair of neighbor nodes n_i and n_j can negotiate a session key as follows:

1. Node n_i selects a random number $a \in \mathbb{Z}_p^*$ and computes $N_i = PK_{n_i}^a$. Node n_i sends N_i and PK_{n_i} to node n_j.
2. Node n_j selects random numbers $b \in \mathbb{Z}_p^*$ and computes $N_j = PK_{n_j}^b$. Node n_j sends N_j and PK_{n_j} to node n_i.
3. Node n_i calculates the session key as follows:

$$S_{n_i n_j} = e\left(SK_{n_i}, N_j \cdot PK_{n_j}^a\right) \tag{3}$$

4. Node n_j calculates the session key as follows:

$$S_{n_j n_i} = e\left(N_i \cdot PK_{n_i}^b, SK_{n_j}\right) \tag{4}$$

Based on the properties of e, we can prove that

$$\begin{aligned} S_{n_i n_j} &= e\left(SK_{n_i}, N_j \cdot PK_{n_j}^{a}\right) \\ &= \mathrm{e}\left(PK_{n_i}^{s}, PK_{n_j}^{b} \cdot PK_{n_j}^{a}\right) \\ &= \mathrm{e}\left(PK_{n_i}^{s}, PK_{n_j}^{a+b}\right) \\ &= e\left(PK_{n_i}, PK_{n_j}\right)^{s(a+b)} = S_{n_j n_i} \end{aligned} \tag{5}$$

At last, nodes n_i and n_j get a session key which can be used to securely transmit data. These session keys are dynamic and related to the nodes' public keys. In this paper, we assume that each pair of node communicate with each other based on the session key and the adversary cannot decrypt the packets in the network.

B. Light-Weight Message Splitting and Sharing Scheme

In this section, we first design a (t, n)-threshold message splitting and sharing approach based on congruence equations and then we prove its correctness and security in Theorem 1 and Theorem 2, respectively. Note that, t and $n(t \leq n)$ are preset by the network operators. For message M generated by a source node, we first encode it by an interleaving coder and then split it into t pieces of sub-messages $x_1, x_2, \cdots, x_t$ with equal lengths. The interleaving coder is employed to destroy the semantic meanings of each single sub-message [38]. For example, if the original message M is "*aaabbbccc*", then the encoded form is "*abcabcabc*". Then, the n shares $s_1, s_2, \ldots, s_n$ are constructed based on $x_1, x_2, \cdots, x_t$ by the following equations where $p = max(x_1, x_2, \cdots, x_t)$.

$$s_i = \begin{cases} x_1 + x_2 + \cdots + x_t \bmod p, & if\ i = 1 \\ s_1 + \cdots + s_{i-1} + x_i + \cdots + x_t \bmod p, & if\ 1 < i \leq t \\ s_1 + 2^{i-t-1} s_2 + \cdots + t^{i-t-1} s_t \bmod p, & if\ t < i \leq n \end{cases} \tag{6}$$

It can be observed from equation (6) that all the shares can be constructed based on $n(t-1)$ additive operations and $(n-t)(t-1)$ multiplicative operations, if all the constant coefficients 2^{i-t-1},3^{i-t-1}, $\cdots$,$t^{i-t-1} (t < i \leq n)$ are pre-calculated and stored. In Shamir's secret sharing scheme, the shares are constructed based on $n(t-1)$ additive operations, $n(t-1)$ multiplicative operations and $n(t-1)$ exponential operations. Therefore, our scheme is of lower computation complexity compared with the classic secret sharing scheme. In addition, considering that $p = max(x_1, x_2, \cdots, x_t)$, the length of the shares in our scheme is about $1/t$ to that of message M. Constructing an anonymity cloud based on the shares is much more energy-efficient than constructing the anonymity cloud based on message M. Overall, the proposed secret sharing scheme is lightweight and it suits WSNs well.

Having obtained n message shares $s_1, s_2, \ldots, s_n$, if the source node can successfully deliver at least t shares to the sink node by employing any existing routing algorithms, the sink node can reconstruct message M based on the received shares. Meanwhile, if the adversary intercepts less than t shares, he cannot recover message M. In the following, we prove the correctness and security of the proposed message sharing scheme in Theorem 1 and Theorem 2, respectively.

Theorem 1 (Correctness): If the sink node receives at least t shares constructed by congruence equations (6), the sink node can reconstruct message M.

Proof: Having received at least t shares, the sink node first needs to construct congruence equation set CE based on any t received shares with the same manner as equation (6). We need to prove that the sink node can get $\{x_1, x_2, \cdots, x_t\}$ by solving CE and further construct M through an interleaving decoder. In other words, we need to prove that equation set CE has a unique solution and it is proved by considering two cases in the following.

Case 1: The t equations that compose equation set CE are the first t equations of the congruence equations (6) and hence CE can be presented as follows:

$$\begin{cases} s_1 = x_1 + x_2 + \cdots + x_t \bmod p \\ s_2 = s_1 + x_2 + \cdots + x_t \bmod p \\ \vdots \\ s_t = s_1 + s_2 + \cdots + s_{t-1} + x_t \bmod p \end{cases} \tag{7}$$

Equivalently, we can express CE in the form of matrix as follows:

$$\begin{aligned} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_t \end{pmatrix} &= \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 2 \\ \vdots & \vdots & & \vdots \\ 2^{t-1} - 2^{t-2} & 2^{t-1} - 2^{t-3} & \cdots & 2^{t-1} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix} \\ &= B \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix} \end{aligned} \tag{8}$$

We can obtain a unit matrix from B by means of simple elementary transformation and thus B is invertible. Further, we know that CE has a unique solution $\{x_1, x_2, \cdots, x_t\}$ and message M can be reconstructed.

Case 2: The first $i (0 \leq i < t)$ equations of CE are chosen from congruence equations in (7) and the other $t - i$ equations of CE are chosen from the last $n - t$ congruence equations in (6) which are presented in the following:

$$\begin{cases} s_{t+1} = s_1 + s_2 + \cdots + s_t \bmod p \\ s_{t+2} = s_1 + 2s_2 + \cdots + ts_t \bmod p \\ \vdots \\ s_n = s_1 + 2^{n-t-1} s_2 + \cdots + t^{n-t-1} s_t \bmod p \end{cases} \tag{9}$$

We now prove that any subset of t shares from $\{s_1, s_2, \ldots, s_t, s_{t+1}, \ldots, s_n\}$ is equivalent to the subset of the first t shares $\{s_1, s_2, \ldots, s_t\}$. Suppose that we choose $i (0 \leq i < t)$ shares: $\{s_{k_1}, s_{k_2}, \cdots, s_{k_i}\}$, $1 \leq k_1 < k_2 < \cdots < k_i \leq t$ from $\{s_1, s_2, \ldots, s_t\}$ and choose another $t - i$ shares: $\{s_{t+k_{i+1}}, s_{t+k_{i+2}}, \cdots, s_{t+k_t}\}$, $1 \leq k_{i+1} < k_{i+2} < \cdots < k_t \leq n - t$ from $\{s_{t+1}, s_{t+2}, \ldots, s_n\}$. In this case, CE can be expressed in the form of matrix as follows:

$$\begin{pmatrix} s_{k_1} \\ \vdots \\ s_{k_i} \\ s_{t+k_{i+1}} \\ \vdots \\ s_{t+k_t} \end{pmatrix} = \begin{pmatrix} 0 \cdots & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 \cdots & 0 & \cdots & 1 & \cdots & 0 \\ 1 \cdots & k_1^{k_{i+1}-1} & \cdots & k_i^{k_{i+1}-1} & \cdots & t^{k_{i+1}-1} \\ \vdots & \vdots & & \vdots & & \vdots \\ 1 \cdots & k_1^{k_t-1} & \cdots & k_i^{k_t-1} & \cdots & t^{k_t-1} \end{pmatrix} \begin{pmatrix} s_1 \\ \vdots \\ s_i \\ s_{i+1} \\ \vdots \\ s_t \end{pmatrix}$$

$$= D \begin{pmatrix} s_1 \\ \vdots \\ s_i \\ s_{i+1} \\ \vdots \\ s_t \end{pmatrix} \tag{10}$$

Now we need to prove that the equations constructed by $\{s_{k_1}, s_{k_2}, \cdots, s_{k_i}, s_{t+k_{i+1}}, \cdots, s_{t+k_t}\}$ is equivalent to those constructed by $\{s_1, s_2, \ldots, s_t\}$. In other words, these two equation sets should have the same solution. We can compute determinant of D and find that $|D| \neq 0$. Therefore, the matrix D is invertible and $\{s_1, s_2, \ldots, s_t\}$ can be linearly expressed by $\{s_{t+k_{i+1}}, s_{t+k_{i+2}}, \cdots, s_{t+k_t}\}$. Based on Case 1, we prove that the equations in Case 2 also have a unique solution.

In fact, we can rewrite the congruence equations (6) in the form of the matrix in the following:

$$\begin{pmatrix} s_1 \\ \vdots \\ s_t \\ s_{t+1} \\ \vdots \\ s_n \end{pmatrix} = H_{n\times t} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{pmatrix} \tag{11}$$

Any subset of t shares from $\{s_1, s_2, \ldots, s_n\}$ corresponds to t rows of the matrix H. By Case 1, we know that the first t shares $\{s_1, s_2, \ldots, s_t\}$ can uniquely decide $\{x_1, x_2, \cdots, x_t\}$. By Case 2, we know that any subset of t shares is equivalent to first t shares $\{s_1, s_2, \ldots, s_t\}$. Combine Case 1 and Case 2, we know that any t rows of the matrix H are linearly independent and any t shares can uniquely decide $\{x_1, x_2, \cdots, x_t\}$. Further, message M can be reconstructed successfully by an interleaving decoder. This completes the proof.

According to Theorem 1, we know that the sink node can reconstruct message M if at least t shares are successfully received. This can significantly improve the robustness of the data transmission process and besides, it can also enhance the security of the transmitted messages in the networks. In tradition, message M is sent to the sink nodes through only one routing path and once an adversary finds this routing path, it can intercept M and begin to decrypt the message. However, in our proposed approach, all the shares are delivered through independent paths and they may be transmitted to different sink nodes. It is extremely difficult for the adversaries to capture a set of shares. Even if an adversary eavesdrops several shares, it is impossible that message M can be reconstructed. In the worst case, i.e., $t1$ shares are collected by the adversaries, we prove that they cannot reconstruct message M even if all the shares are decrypted successfully in Theorem 2.

Theorem 2 (Security): If the adversary intercepts less than t shares of message M, he cannot reconstruct message M accurately.

Proof: We first assume that the strong adversary has intercepted $t-1$ shares and all the intercepted shares have been successfully decrypted. Then, to reconstruct message M, the adversary needs to solve the equation set that is constructed by the $t-1$ shares. Apparently, if the adversary cannot reconstruct message M from the $t-1$ shares, he cannot reconstruct message M from a set of shares with less than $t-1$ shares. Let F be a field and $H = (h_{ij})_{n\times t} = (H_1, H_2, \ldots, H_n)^T$ be a matrix over F. Given n numbers $s_1, s_2, \ldots, s_n$ from the field F, consider a matrix equation:

$$H_{n\times t} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_t \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_n \end{pmatrix} = S \Leftrightarrow HX = S, \ where \ X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_t \end{pmatrix} \tag{12}$$

Consider the worst-case scenario: the matrix H has a rank $t-1$ and we need to show the matrix equation has no solution or has $|F|$ solutions.

Consider matrix H and let

$$\begin{pmatrix} H_1 & s_1 \\ H_2 & s_2 \\ H_3 & s_3 \\ \vdots & \vdots \\ H_n & s_n \end{pmatrix} = \overline{H} \tag{13}$$

Case 1: If the rank of H is not equal to the rank of $\bar{H}$, the matrix equation has no solution and in this case, the adversaries cannot recover message M.

Case 2: If the rank of H is equal to the rank of $\bar{H}$, the matrix equation has a special solution:

$$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_t \end{pmatrix} = A, \quad HA = S \tag{14}$$

and the matrix equation $HX = 0$ has a solution vector space of 1-dimension over F generated by vector Y. Hence it has $|F|$ solutions: $kY (k \in F)$. Hence, $HX = S$ has solutions: $kY + A (k \in F)$. Combining Case 1 and Case 2, Theorem 2 is proved.

Based on Theorem 2, we can observe that even the adversary captures some shares of message M, he cannot recover M. Consequently, the message sharing scheme improves the confidentiality of data in the network.

When executing the scheme with pre-selected parameters t and n, we can pre-compute the coefficient matrix H in (11) and preload them into the sensor nodes before deploying the networks. Then, there are only add and multiply operations both of which are very energy-efficient compared with most

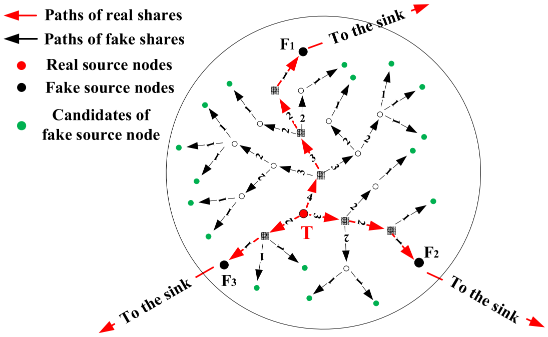

Fig. 3. Spreading the anonymity cloud.

existing message sharing schemes. Overall, this scheme performs very well in terms of energy efficiency in the whole process and as a result, it is an extremely suitable message sharing scheme for WSNs.

C. Anonymity Cloud Construction Based on Message Shares

1) Spreading the Anonymity Cloud: To protect source-location privacy, the source node uses the shares $s_1, s_2, \ldots, s_n$ to construct an anonymity cloud of an irregular shape in the around area. In this way, the adversary cannot locate the source node by analyzing the shape of the cloud. In our scheme, the size of an anonymity cloud is defined as the number of nodes covered by the cloud and it is indirectly decided by the number of average hops h that the shares can be transmitted in the cloud. According to different security demands, parameter h needs to be preset by the operators. For a network with a high-security requirement, the size of the anonymity cloud should be increased and on the contrary, if the network has a low-security requirement, the size of the anonymity cloud should be decreased. In theory, if we increase h of a cloud by α times, the size of the cloud would increase by α^2 times. As a result, the total energy consumption of constructing the cloud also increases by α^2 times. Therefore, the operators of the networks should select parameter h carefully to find a good balance between security and energy consumption.

We assume that each share s_i has a random hop count h_i which is generated by a uniformly random distribution with mean value h. In this paper, h_i is generated by a uniform distribution $U(0.1h, 1.9h)$ whose mean value is h. Though h is a constant number for an anonymity cloud, the shape of the anonymity cloud is random and irregular. This can be explained by the fact that some shares' hop counts are larger than h and some other shares' hop counts are smaller than h. A brief example of constructing an anonymity cloud is shown in Fig. 3. Initially, source node T generates message M and 3 message shares are constructed by the message sharing scheme presented in Section IV.B. We assume that $h = 3$ and the corresponding hop counts for the three shares are 2, 3 and 4, respectively. Then the source node sends the three shares to its neighbors randomly and the real shares are delivered in the network along with the three red paths. Meanwhile, some fake shares are also generated and broadcasted in the network to protect the real shares from being identified by the adversary. The cloud stops spreading once the hop counts of the shares decrease to 0 and the corresponding nodes, i.e., F_1, F_2, F_3, that receive the real shares are defined as the fake source nodes. Meanwhile, all the nodes that receive either real shares or fake shares with hop counts 0 are defined as candidates of fake source nodes and they may become a fake source node for the next share. At last, the fake source nodes send the real shares to the sink node through proper routing algorithms immediately, which will be discussed in the next section. It can be observed that the cloud is constructed in a totally distributed manner and the fake source nodes are selected in a random way which increases the difficulty of back-tracing.

Algorithm 1 ConstructionOfAnonymityCloud

```
1: An event is monitored by the source node $T$ and a
   message $M$ with identifier $ID_M$ needs to be sent to the
   sink node;
2: Node $T$ generates $n$ shares $s_1, s_2, \cdots, s_n$ of message $M$
   and they share the same identifier $ID_M$;
3: Node $T$ randomly selects a hop count $h_i$ for each share
   $s_i$
4: for each node $N_j$ receiving a share $s_i$
5:   $N_j$ decrypts $s_i$ and checks whether $s_i$ is a real share;
6:   if $s_i$ is a real share and $h_i$ is zero
7:     $N_j$ send the share to the sink node immediately;
8:   else if $s_i$ is a real share and $h_i$ is not zero
9:     $N_j$ updates $h_i$ by $h_i - 1$ and randomly sends $s_i$ to a
       neighbor node with a random delay $t_{real}$;
10:    $N_j$ generates several fake shares which have the same
       hop count and identifier with $s_i$ and send them to the
       other neighbors with a random delay $t_{fake}$;
11:  else if $s_i$ is fake and $h_i$ is not zero and $N_j$ has never
       received other shares with the same identifier
12:      $N_j$ updates $h_i$ by $h_i - 1$ and sends the fake shares
       to all its neighbors with a random delay $t_{fake}$;
13:  else
14:    $N_j$ drops $s_i$;
15:  end if
16: end for
```

The detailed process of anonymity cloud construction is presented in Algorithm 1. We assume that message M has a unique identifier and the identifier is added into the heads of all its shares including real and fake shares. For each node n_i that receives a share s_i, it starts to check whether the share is a real share. As shown in line 6 to 10, if the share is real and its hop count is zero, node n_i becomes a fake source node and it sends the share to the nearest sink node immediately; if the share is real and the hop count is not zero, node n_i needs to subtract the hop count by one and then send the share to one of its neighbors with a time delay t_{real}. Meanwhile, some fake shares with the same identifier are generated and sent to the other neighbors of n_i. As shown in line 11 to 15, if node n_i receives a fake share and has never received a fake share with the same identifier, it subtracts the hop count by one and then

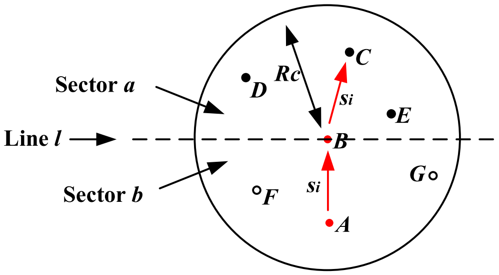

Fig. 4. Selection of the real share's next hop.

sends the fake share to all its neighbors with a random time delay t_{fake}; if n_i has received a fake share with the same identifier, it just drops the share. Thus, fake shares are not transmitted repeatedly in the cloud. We will discuss how to choose the next hop of s_i and generate t_{real} and t_{fake} in the following.

The shape of the cloud performs an important role in protecting source location and it is decided by the hop counts of the shares. Note that, once a cloud is constructed, its shape keeps stable and hence the candidates of fake shares are constant. When a new message is generated by a node in the cloud, the node needs not to generate hop counts for the shares and instead the shares just need to be delivered in the cloud until they reach a candidate of fake source node. At last, the candidate fake node becomes a fake source node and it sends the share to sink node.

2) Choosing the Next Hop of a Real Share: We now discuss how to choose the next hop when delivering a real share in the cloud. If the nodes send a real share to a neighbor node in a totally random manner, it is likely that the real share is transmitted around the real source node repeatedly and the walk steps conceal with one another. This is a bad phenomenon for protecting the source location considering that the cloud cannot spread in time and it also increases the workload of the sensor nodes around the source node. In this paper, we design a new strategy to choose the next hop of a real share based on the sector-based directed walk model. As shown in Fig. 4, a real share s_i is sent from node A to node B and node B needs to select the next hop of s_i. We assume that the communication radius of node B is R_c and it has 6 neighbors namely $A, C, D, E, F,$ and G. Apparently, it is unacceptable that node B sends s_i back to node A and hence the choice of the next hop must be one of the nodes in $C, D, E, F,$ and G. Considering that we do not want the real shares to be sent back and forth, we divide the whole communication range of node B into two half-circles, i.e., sector a and sector b, based on line l which is perpendicular to line AB and goes through node B. Then, only the nodes in Sector a, i.e., C, D, and E, are legal candidates of the next hop and the other nodes, i.e., F and G, are illegal candidates. Node B can randomly choose a legal candidate as the next hop of share s_i. In Fig. 4, node B selects node C as the next hop of s_i. However, in some cases, node B may have no legal neighbors in its communication range, in which case node B needs to choose the neighbor nearest to line l as the next hop.

3) Generating Time Delays for the Shares: Another challenge is to select proper time delays for both the fake and real shares. In a cloud, the fake shares contain no valuable information and they are employed to hide the real shares. When delivering fake shares in the cloud, a randomly generated time delay $t_{fake} \sim N(\mu, \sigma^2)$ is employed by all the nodes to destroy the regular time patterns beneath the shares. If t_{fake} is too small, the time pattern cannot be destroyed thoroughly, because the shares of different messages cannot coexist in the same cloud and it is easy for the adversaries to analyze the orders of the nodes in the process of transmitting shares. If t_{fake} is too large, the freshness of message M decreases which is unacceptable for the data users. Overall, we need to set t_{fake} in a proper way to achieve a balance between security of the source node and timeliness of the data. We assume that the source node generates messages with a constant frequency f and hence the period is $1/f$. In this paper, we set $\mu = 1/f$ and $\sigma = 1/3f$ which is an extension of the distribution of the real shares. In this case, we can hide the real shares in the fake shares and they are generated with the same frequency in the cloud. In fact, the radio actions of the nodes in the cloud are similar to that of the nodes in the global-adversary-based source-location privacy protection schemes.

Different from the fake shares, the real shares carry the information about the monitored targets and their timeliness should be improved compared with that of the fake shares. To decrease time delay, the best method is transmitting the real shares immediately once they are received by a set of nodes. However, the time delays of fake shares in the cloud follow an approximate constant distribution and sending a real share without a proper time delay can be detected by the adversary through analyzing the radio actions of a node [14]. For example, the adversary can observe radio transmissions of a set of nodes over an extended period of time and perform sophisticated statistical analysis to compare the observed data transmission pattern with the known distribution of fake shares. If the observed pattern cannot match the traditional models properly, the adversary would suspect that some real shares are transmitted by the nodes recently. However, if t_{real} is too large, the total time delay of delivering message M from the source node to the sink node increases and it decreases the timeliness of the monitored data. Therefore, in this paper, we employ the approach in [14] to set minimal time delays for the real shares. It has been detailedly analyzed in [14] that the adversary cannot distinguish whether some real shares are sent by a node in a period of time by employing specific fit test models including the AndersonDarling (A-D) test [39], the Kolmogorov-Smirnov (K-S) test [40] and the Jarque-Bera (J-B) test [41]. By doing this, the adversary cannot find particular statistical rules of time delays in the cloud and all the nodes in the anonymity cloud are indistinguishable no matter whether they have sent some real shares in a period of time.

4) Updating and Merging the Anonymity Cloud: Though the cloud keeps stable in general, the parameter h_i of the shares need to be updated if the monitored target stays in a field for a long time and hence the cloud will be updated. The process of constructing a new cloud is straightforward as discussed previously.

When a new source node is generated near to an existing cloud, the new constructed cloud may intersect to the old one. In this case, we need to merge them to a larger anonymity cloud. Specifically, if a sensor node receives multiple fake shares from different clouds, it sends just one fake share that has the largest number of hop counts and drops the other fake shares. However, if several real shares are received by a candidate fake source node, the fake source node sends them to the sink node in order with proper time delays. Note that, the outside shares cannot be delivered to a cloud without the help of border nodes, i.e., the candidates of fake source node, considering the cloud construction process presented in Algorithm 1.

All the shares generated from a message M are sent out by the source node at one time and hence only one anonymity cloud is constructed for message M. We define the whole process of delivering a message from the source node to the sink nodes based on message sharing scheme as a message delivery round. In average, each node in the cloud needs to transmit less than one fake share in a round and some nodes may transmit at least a real share with an extremely low probability. Overall, each node in the cloud needs to transmit about one packet with the same length of real shares. As a consequence, the nodes in our scheme are much more energy-efficient than the nodes in the existing cloud-based schemes.

D. Message Delivery to the Sink Node and Reconstructing Message M

As discussed in Section IV.C, the fake source nodes send the shares to the sink node. Any existing routing algorithm can be employed to deliver the shares including both constant routing algorithms and random routing algorithms. Intuitively, random routing algorithms (e.g., Phantom routing algorithm) can be seamlessly integrated into our scheme and they can make the proposed approach perform better in protecting the source location privacy. This is reasonable considering that random routing algorithms can further disperse the routing paths and improve the difficulty of back-tracing. In addition, to make a message indistinguishable in the routing path, we can employ the pseudonym technique in [11]. However, in this paper, we focus our attention on the anonymity-cloud-based source-location protection method. Though several shares are sent to the sink node, the total data transmission amount does not increase significantly considering that the length of the shares is much shorter than that of message M.

Once the sink node receives at least t shares, it can reconstruct the original message M based on the shares received. For each share s_i, the sink node can construct an equation as follows:

$$s_i = \begin{cases} \sum_{j=1}^{t-1} x_j \left(\sum_{i=1}^{t} 2^{i-1} - 2^{i-j-1}\right) + 2^{t-1} x_t \bmod p, \\ \qquad if\ 1 \leq i \leq t \\ \sum_{j=1}^{t} x_j \left(\sum_{\upsilon=1}^{t} \upsilon^{i-t-1} - j^{i-t-1}\right) \bmod p, \\ \qquad if\ t < i \leq n \end{cases} \tag{15}$$

which is equivalent to equation (6).

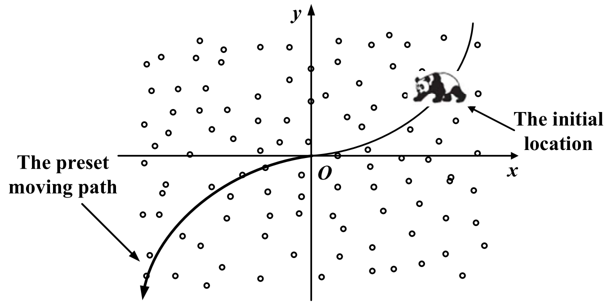

Fig. 5. The motion model of the hotspot.

Considering that any $s_i (1 \leq i \leq n)$ can be linearly expressed by $\{x_1, x_2, \ldots, x_t\}$, we can get t linearly independent equations with $\{x_1, x_2, \ldots, x_t\}$ as unknown variables. Based on Theorem 1, we can solve the equations by the Gauss Elimination Method for a unique result of $\{x_1, x_2, \ldots, x_t\}$. At last, message M can be reconstructed through an interleaving decoder based on $\{x_1, x_2, \ldots, x_t\}$ and then the message delivery process is completed. As discussed in Section III, we assume that the sink node is of sufficient power and hence the energy consumption of reconstructing message M is ignored in our scheme.

V. Performance Evaluation

A. Simulation Settings

In this section, we evaluate the performance of SPAC on packet delivery layer based on ns-3 discrete event simulator (version ns-3.21). In our simulation, 6000 sensor nodes are scattered in a 4000 m × 4000 m square region. The sink node locates in the center of the network and the farthest distance between a source node and the sink node is about 40 hops. We construct the routing paths between fake source nodes and the sink node based on geographic information of the nodes. This is reasonable considering that geographic routing algorithms are of great scalability and do not strictly limit the hop counts in routing process. Therefore, they suit large WSNs very well. For convenience sake, we assume that only one panda exists in the network. The motion model of the panda is defined as follows. First, a preset moving path is generated by a cubic polynomial and the moving speed is set to be 1 m/s. Specifically, we build a coordinate system with the original point at the center of the network. Then, we randomly generate three numbers to act as the coefficients of the cubic polynomial and employ the shape of the polynomial in the coordinate system as the preset path. At last, the initial location of the monitored hotspot is randomly chosen on the path and the panda moves to the direction of the initial point. The preset path generated by polynomial y $=x^3$ is shown in Fig. 5 and the panda moves from the top right corner to the bottom left corner of the network.

In the simulation, the network employs the 3-nearest neighbors tracing approach [37] to monitor the target and the source node sends the collected information to sink nodes once a target is monitored. In the process of monitoring, the generated messages are transmitted to the sink nodes continuously with a period of 1 second. Each message M is 1024 bits and

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Size of network	4000 m × 4000 m
Number of nodes	6000
Number of target	1
Number of adversaries	1
(t, n)	(4,7)
N_a	$(40, 80, \cdots, 320)$
N_p	$(4, 8, \cdots, 32)$
R_c	80 m
R_o	80 m
Target monitoring scheme	3-nearest neighbors tracing scheme
Package transmission interval	1 s
Length of message M	1024 bits
Length of the package head	32 bits

the head of each package is 32 bits. In the message sharing scheme, t and n are set to be 4 and 7, respectively. Therefore, 7 shares are generated and any 4 of them can recover the original message M. The size of the anonymity clouds N_a is selected from the set $(40, 80, \ldots, 320)$. In addition, $N_p \in (4, 8, \ldots, 32)$ parasitic nodes are initially deployed around the sink nodes and they try to trace back to the source nodes.

The simulation parameters are summarized in Table 1. In this paper, we mainly compare the performance of SPAC with that of the shortest path routing algorithm, phantom routing algorithm and the cloud-based scheme [11]. In the shortest path routing algorithm, the source node always selects the nearest sink node as the destination of the messages and ignores the back-tracing threat. In phantom routing algorithm, the number of random walk steps is set to be equal to the average hop count of the shares h which is related to the size of the anonymity cloud N_a. After the random walk phase, the fake source node sends the message to the nearest sink node. The cloud-based scheme shares the same parameters with SPAC and it is the most important benchmark to be compared with SPAC. Each simulation is operated for 100 times independently. We terminate a simulation if the distance between a parasitic node and the source node is smaller than R_c or the simulation lasts for 1000 seconds.

B. Source Location Security With Different Number of Parasitic Nodes

We apply the source detection probability to evaluate the performance of SPAC in terms of privacy preservation. The source detection probability is defined as the probability that the parasitic nodes can locate the source nodes successfully. In our simulation, it is measured by the number of times that the parasitic nodes locate the source nodes, relation to the total number of the simulation runs. With different number of parasitic nodes, simulation results are presented in Fig. 6. The size of anonymity cloud N_a is set as 160 and we observe that with the increasing of parasitic nodes' number, the source detection probabilities increase for all the four approaches. The shortest path routing algorithm cannot provide any protection on the source-location privacy, because it always chooses similar routing paths for the same source node and sink node. As a result, it is very easy for the adversaries to trace back to the source node. When $N_p = 32$, i.e., the adversaries deploy 32 parasitic nodes in the network, they can find the source node with a probability higher than 90%. Apparently, this is unacceptable for most network users. Phantom random routing algorithm always outperforms the shortest path routing algorithm. This can be explained by the fact that a random walk phase is employed to destroy the constant routing paths which can confuse the adversaries to some extent. However, when $N_p = 32$, the adversaries can find the source node with a probability of about 45%, which is also hard to satisfy the network users.

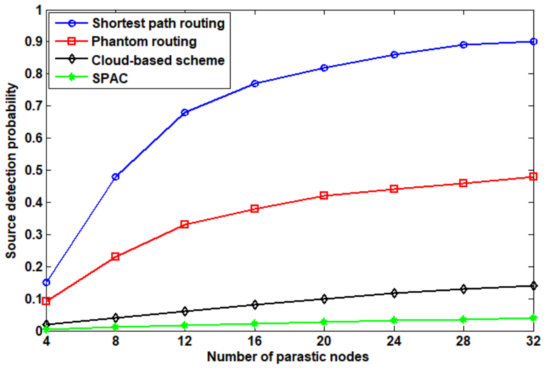

Fig. 6. Source detection probability with different number of parasitic nodes.

Compared with routing-based approaches, SPAC and cloud-based scheme perform much better in protecting source-location privacy. Though the methods of constructing the anonymity clouds are totally different, both of the two clouds can hide the real source node among a set of members. With the increasing of the parasitic nodes' number, the source detection probability increases slowly. As shown in Fig. 6, SPAC outperforms the cloud-based scheme and it is difficult for the adversary to locate the source node accurately. Specifically, source detection probability of SPAC is about 30% of that in the cloud-based scheme. This can be explained by the fact, the radio actions of the nodes in the anonymity cloud is carefully designed and that in cloud-based scheme is not considered. When $N_p = 32$, the adversaries can find the source node with a probability of about 4% in SAPC which is much smaller than that of routing-based approaches and the cloud-based scheme.

C. Source Location Security With Different Size of the Cloud

To further compare the two cloud-based approaches, we present source detection probability with different sizes of clouds. In this simulation, we set N_p as 16 and change the size of cloud from 40 to 320. Apparently, the size of the

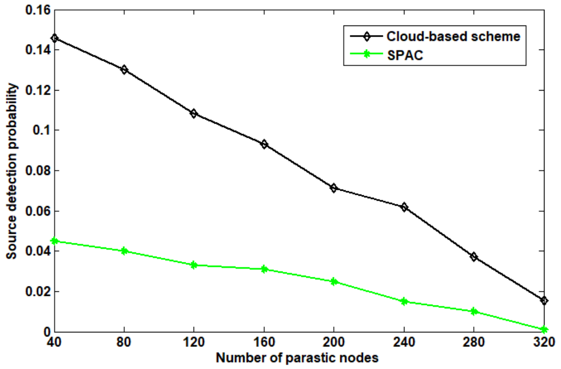

Fig. 7. Source detection probability with different size of the cloud.

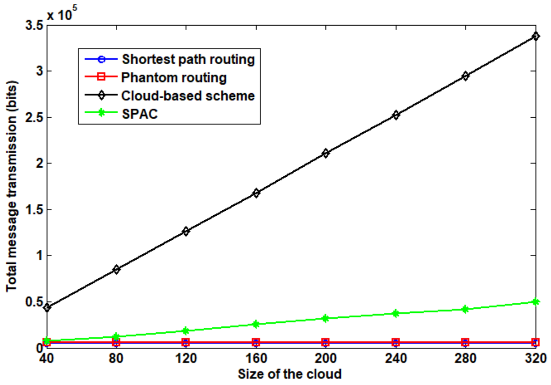

Fig. 8. Total message transmission with different size of the cloud.

cloud has no effect on the routing-based schemes including shortest path routing algorithm and phantom routing algorithm. With the same parameter, the source detection probabilities for the routing-based approaches are 0.77 and 0.38 respectively which are much larger compared with that of cloud-based schemes. For simplicity, we present only the simulation results of the two cloud-based approaches and ignore that of the routing-based schemes. As shown in Fig. 7, with the increasing of the cloud size from 40 to 320, the source detection probability decreases significantly for both of the two cloud-based approaches. However, SPAC performs much better than the cloud-based scheme in protecting the source locations. This can be explained by the fact that the radio behaviors of the nodes in our scheme are carefully designed.

D. Data Transmission Amount

The data transmission amounts of source-location privacy protection approaches are greatly affected by the sizes of clouds. As shown in Fig. 8, the total data transmission amounts of routing-based schemes are not affected by the size of the cloud. However, data transmission amount of SPAC and the cloud-based scheme increase significantly with the increasing of the cloud size. SPAC and cloud-based scheme transmit more messages than routing-based approaches, because many redundant messages are transmitted in the cloud to protect the real messages. Considering that the anonymity clouds cover a large number of nodes and the routing paths cover just several nodes, the performance gaps between SPAC, the cloud-based scheme and the other two routing-based schemes are very large. However, SPAC performs much better than the cloud-based scheme which can be explained by the fact that we employ shares to construct the anonymity cloud rather than the original message M. Besides, the proper time delay of the fake shares also decreases the total data transmission amount.

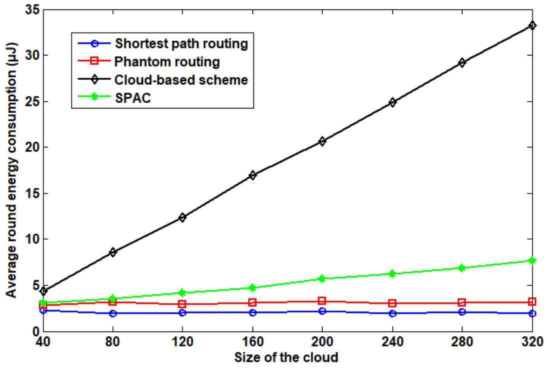

Fig. 9. Average round energy consumption with different size of the cloud.

E. Energy Consumption

Though most energy is consumed in the process of sending and receiving messages in WSNs, a large amount of energy is consumed in the computing process. In this section, the average energy consumptions per round of the schemes are discussed. Note that, the energy consumption of the sink node is not reflected in simulation results considering that the sink node has enough power.

In this simulation, we employ the radio energy dissipation model proposed in [42] and the computing energy consumption model proposed in [43]. For each round of message delivery, the average energy consumption is presented in Fig. 9. Similar to the data transmission amount, the energy consumptions of routing-based schemes are stable with the increasing of the cloud size and that of SPAC and the cloud-based scheme monotonously increase with the increasing of the cloud size. However, our scheme performs much better than the cloud-based scheme and this shares the same reason with that of data transmission amount.

F. Reliability of Data Transmission in the Network

Besides providing strong source-location privacy protection with low energy consumption, our approach provides reliable data transmission between the source nodes and the sink node and hence SPAC greatly improves the robustness of the networks. We use success rate of receiving message M by the sink node to illustrate the robustness of our approach. In SPAC, M is broken into n shares and the sink node does not receive M directly. Instead, if at least t shares are received, M is recovered by the sink node and this corresponds to receiving M in our simulation. In the other three approaches, the definition of receiving M is unambiguous. The success rate is measured by the number of times that the sink node receives M to the total number of messages sent by the source nodes. The simulation

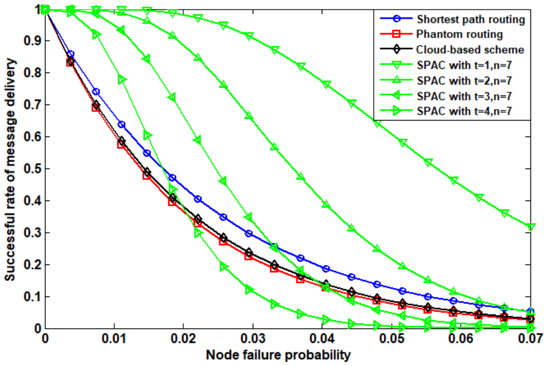

Fig. 10. The probability of receiving message M by the sink node with different failure probabilities of the sensor nodes.

results with different average failure probabilities of the sensor nodes are presented in Fig. 10.

It can be observed that the reliability of the nodes has significant effect on the success rates of message delivery. All the three existing approaches including the shortest path routing, Phantom routing and cloud-based scheme have poor performances on providing reliable data transmission. In these three approaches, even the average failure probability of the nodes is only 0.01, the link between a source node and a sink node fails with a probability of about 35%. With the increasing of the average failure probability, the success rate of delivery of messages decreases rapidly. When the average failure probability increases to 0.018, more than 50% messages are dropped in the delivery process and this is unacceptable. In theory, the success rate of message delivery is strongly related to the average hop counts between the source node and sink node. We can find that the cloud-based scheme and Phantom routing approach share similar performance because of their similar average hop counts of packages. The shortest routing approach performs some better because the shortest routing approach has the smallest average hop counts.

SPAC has a high fault tolerance on the failure of nodes, because it employs the message sharing scheme with proper parameters. In simulation, we set $n = 7$ and t is chosen from the set of $\{1, 2, 3, 4\}$. For all the combinations of parameters, SPAC outperforms the other three approaches and the success rates are all larger than 99% when the node failure probability is 0.005. When we set $t = 1$, the sink node almost always receives message M with node failure probability ranging from 0 to 0.017. This is reasonable considering that the sink node needs to receive only one of seven shares successfully. For each t, the success rate decreases with the increasing of node failure probability, and the larger t turns, the faster the success rate decreases. For each network, the operator needs to set the parameters properly based on the quality of the sensor nodes and reliability requirements of the network. For example, if node failure probability is 0.005 and the user requires a success rate higher than 95%, t can be selected from $\{1, 2, 3, 4\}$; however if node failure probability is 0.02 and the user requires a success rate higher than 90%, t can be selected from $\{1, 2\}$.

G. Performance Discussion

Through a series of simulations, we can observe that the routing-based approaches cannot provide strong protection to the source-location privacy under the enhanced Hotspot-Locating attack. The shortest path routing algorithm is the most vulnerable approach because the paths of the messages are always very similar with each other. Though this approach is greatly energy-efficient, it is useless in protecting source-location privacy. The Phantom routing algorithm adds a random walk phase into the routing process. Apparently, this makes the routing paths diverse with each other even the source node and the sink node are constant. However, it is also likely for the adversaries to trace back to the source node. When a stream is continuously transmitted to a sink node, the parasitic nodes can first trace back to the fake sources which are about h hops away from the source node. Considering that the adversary can deploy all its sources in the suspected region, he can easily locate the source node. Though some extra energy is consumed in the random walk phase compared with the shortest path routing algorithm, the Phantom algorithm is still very energy-efficient. Phantom routing algorithm provides weak protection on the source-location privacy with a very slight increasing of energy consumption. For the networks with extremely limited sources and very weak security requirements, it is a good choice to employ the Phantom routing algorithm. However, for networks with high safety requirements, routing-based approaches are not sufficient.

Compared with the routing-based approaches, SPAC and the cloud-based scheme can provide much stronger protection on the source locations. For the cloud in [11], only one message is valuable and all the other messages are fake ones. Most energy is consumed in the transmission of fake messages, particularly when the cloud is enormously large. The greatest disadvantage of this approach is its low energy-efficiency. Before constructing the cloud, a Bootstrapping phase is needed to construct the groups and the fake source nodes. Once the topology of the network changes, another Bootstrapping phase needs to be re-executed, which significantly decreases the dynamic of the approach. Moreover, the authors have not yet designed a method properly to set the random delays when transmitting messages in the cloud and it is possible for the adversary to locate the source location accurately once he finds the boundary of the cloud. In summary, the cloud-based scheme is a qualified approach in protecting source location privacy, but there is still room for improvement in terms of both source-location protection effect and efficiency. SPAC is a totally distributed approach and it is unnecessary to consider the change of network topology. For each source node, it needs not to decide the fake source nodes in advance and it just needs to send the packages to its neighbors. In addition, our random delay mechanism completely destroys the time patterns beneath the packages in the cloud. This makes it almost impossible for the adversaries to locate the source node accurately. As for fault tolerance, our scheme outperforms the other three schemes significantly because of the employing of message sharing scheme. In summary, for the networks with limited resources and high-security requirements, the best choice is employing SPAC to improve network security.

VI. Conclusion and Future Work

In this paper, we propose a novel anonymity-cloud-based source-location privacy protection approach to defend against the enhanced Hotspot-Locating attack. We first design a light-weight message splitting and sharing scheme particularly for WSNs based on congruence equations. The correctness and security of the scheme are proved in theorems. Based on the shares, an anonymity cloud is constructed to hide the source node in which all the nodes are indistinguishable. At the boundary of the cloud, some fake source nodes send the shares to the sink node. Lastly, the original message is reconstructed by the sink node based on the received shares. We evaluate the performance of the proposed approach through a series of experiments and compare it with existing source-location privacy protection approaches such as routing-based schemes and cloud-based schemes. Simulation results show that the proposed approach provides strong privacy protection with an energy-efficient manner. In addition, data confidentiality in the network and fault tolerance for the failure of sensor nodes is also greatly improved.

In terms of our future work, we plan to design a two-fold source-location privacy protection scheme in which an anonymity cloud is first constructed around the source node to hide the source node and then an all-direction random routing algorithm is designed based on the geographic information to send the shares from the fake source nodes to the sink nodes. This approach is a combination of routing-based schemes and cloud-based schemes and it may provide all-around protection to source-location privacy in WSNs.

References

[1] Y. Liu, M. Dong, K. Ota, and A. Liu, "ActiveTrust: Secure and trustable routing in wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 2013–2027, Sep. 2016.

[2] C. Zhu, H. Nicanfar, V. C. M. Leung, and L. T. Yang, "An authenticated trust and reputation calculation and management system for cloud and sensor networks integration," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 1, pp. 118–131, Jan. 2015.

[3] S. He, J. Chen, F. Jiang, D. K. Y. Yau, G. Xing, and Y. Sun, "Energy provisioning in wireless rechargeable sensor networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 10, pp. 1931–1942, Oct. 2013.

[4] Z. Zheng, A. Liu, L. X. Cai, Z. Chen, and X. S. Shen, "Energy and memory efficient clone detection in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 5, pp. 1130–1143, May 2016.

[5] Y. Fan, Y. Jiang, H. Zhu, and X. S. Shen, "An efficient privacy-preserving scheme against traffic analysis attacks in network coding," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 2213–2221.

[6] W. Lou, W. Liu, and Y. Fang, "SPREAD: Enhancing data confidentiality in mobile ad hoc networks," in *Proc. INFOCOM*, vol. 4., Mar. 2004, pp. 2404–2413.

[7] S. Ozdemir and H. Cam, "Integration of false data detection with data aggregation and confidential transmission in wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 736–749, Jun. 2010.

[8] M. E. A. Mahmoud, X. Lin, and X. Shen, "Secure and reliable routing protocols for heterogeneous multihop wireless networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 1140–1153, Apr. 2015.

[9] D. Qin, S. Yang, S. Jia, Y. Zhang, J. Ma, and Q. Ding, "Research on trust sensing based secure routing mechanism for wireless sensor network," *IEEE Access*, vol. 5, pp. 9599–9609, 2017.

[10] A. Liu, Z. Zheng, C. Zhang, Z. Chen, and X. Shen, "Secure and energy-efficient disjoint multipath routing for WSNs," *IEEE Trans. Veh. Technol.*, vol. 61, no. 7, pp. 3255–3265, Sep. 2012.

[11] 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 Trans. Parallel Distrib. Syst.*, vol. 23, no. 10, pp. 1805–1818, Oct. 2012.

[12] Y. Zhang, W. Liu, Y. Fang, and D. Wu, "Secure localization and authentication in ultra-wideband sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 4, pp. 829–835, Apr. 2006.

[13] (2012). *WWWF—The Conservation Organization*. [Online]. Available: http://wwf.panda.org/

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

[15] M. Shao, Y. Yang, S. Zhu, and G. Cao, "Towards statistically strong source anonymity for sensor networks," in *Proc. 27th IEEE Conf. Comput. Commun.*, Apr. 2008, pp. 51–55.

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

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

[18] N. Wang and J. Zeng, "All-direction random routing for source-location privacy protecting against parasitic sensor networks," *Sensors*, vol. 17, no. 3, p. 614, 2017.

[19] C. Erbas, M. M. Tanik, and Z. Aliyazicioglu, "Linear congruence equations for the solutions of the N-Queens problem," *Inf. Process. Lett.*, vol. 41, no. 6, pp. 301–306, Apr. 1992.

[20] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao, "Towards event source unobservability with minimum network traffic in sensor networks," in *Proc. ACM Conf. Wireless Netw. Secur.*, Apr. 2008, pp. 77–88.

[21] R. Lu, X. Li, H. Zhu, and X. Shen, "TESP2: Timed efficient source privacy preservation scheme for wireless sensor networks," in *Proc. IEEE Int. Conf. Commun.*, May 2010, pp. 1–6.

[22] K. Mehta, D. Liu, and M. Wright, "Protecting location privacy in sensor networks against a global eavesdropper," *IEEE Trans. Mobile Comput.*, vol. 11, no. 2, pp. 320–336, Feb. 2012.

[23] K. Ren, W. Lou, and Y. Zhang, "LEDS: Providing location-aware end-to-end data security in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 5, pp. 585–598, May 2008.

[24] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[25] P. N. Smart, "Secret sharing schemes," in *Cryptography Made Siple*. Jan. 2016, pp. 403–416.

[26] S. Zou, Y. Liang, L. Lai, and S. Shamai (Shitz), "An information theoretic approach to secret sharing," *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3121–3136, Jun. 2015.

[27] Y. Wei, P. Zhong, and G. Xiong, "A multi-stage secret sharing scheme with general access structures," in *Proc. IEEE 4th Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Oct. 2008, pp. 1–4.

[28] T.-Y. Lin and T.-C. Wu, "(t,n) threshold verifiable multisecret sharing scheme based on factorisation intractability and discrete logarithm modulo a composite problems," *IEE Proc.-Comput. Digit. Techn.*, vol. 146, no. 5, pp. 264–268, Sep. 1999.

[29] D. R. Stinson and S. A. Vanstone, "A combinatorial approach to threshold schemes," *SIAM J. Discrete Math.*, vol. 1, no. 2, pp. 230–236, May 1988.

[30] L.-J. Pang and Y.-M. Wang, "A new (t, n) multi-secret sharing scheme based on Shamir's secret sharing," *Appl. Math. Comput.*, vol. 167, no. 2, pp. 840–848, Aug. 2005.

[31] O. R. Marek and O. Urszula, "The use of mathematical linguistic methods in creating secret sharing threshold algorithms," *Comput. Math. Appl.*, vol. 60, no. 2, pp. 267–271, Jul. 2010.

[32] J. Zhang and F. Zhang, "Information-theoretical secure verifiable secret sharing with vector space access structures over bilinear groups and its applications," *Future Gener. Comput. Syst.*, vol. 52, pp. 109–115, Nov. 2015.

[33] W. Huang, M. Langberg, J. Kliewer, and J. Bruck, "Communication efficient secret sharing," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 7195–7206, Dec. 2016.

[34] L. Harn and M. Fuyou, "Multilevel threshold secret sharing based on the Chinese Remainder Theorem," *Inf. Process. Lett.*, vol. 114, no. 9, pp. 504–509, Sep. 2014.

[35] X. Cheng, A. Thaele, G. Xue, and D. Chen, "TPS: A time-based positioning scheme for outdoor wireless sensor networks," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 2685–2696.

[36] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 6, no. 2, pp. 28–36, Apr. 2002.

[37] Y. Liu, J.-S. Fu, and Z. Zhang, "K-nearest neighbors tracking in wireless sensor networks with coverage holes," *Pers. Ubiquitous Comput.*, vol. 20, no. 3, pp. 431–446, Jun. 2016.

[38] B. Vucetic and J. Yuan, *Turbo Codes: Principles and Applications*. Norwell, MA, USA: Kluwer, 2000.

[39] T. W. Anderson and D. A. Darling, "Asymptotic theory of certain "Goodness of Fit" criteria based on stochastic processes," *Ann. Math. Statist.*, vol. 23, no. 2, pp. 193–212, Jun. 1952.

[40] F. J. Massey, Jr., "The Kolmogorov–Smirnov test for goodness of fit," *J. Amer. Statist. Assoc.*, vol. 46, no. 253, pp. 68–78, 1951.

[41] C. M. Jarque and A. K. Bera, "A test for normality of observations and regression residuals," *Int. Statist. Rev.*, vol. 55, no. 2, pp. 163–172, 1987.

[42] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[43] S. Nithyakalyani and S. S. Kumar, "Data aggregation in wireless sensor network using node clustering algorithms—A comparative study," in *Proc. IEEE Conf. Inf. Commun. Technol. (ICT)*, Apr. 2013, pp. 508–513.

Na Wang received the Ph.D. degree from the School of Mathematical Sciences, Xiamen University, in 2018. She is currently a Post-Doctoral Fellow with the School of Computer Science, Beijing University of Posts and Telecommunications. Her research interests include cryptography, message sharing, and information security issues in distributed and cloud systems.

Junsong Fu received the Ph.D. degree in communication and information system from Beijing Jiaotong University in 2018. He is currently an Assistant Professor with the School of Cyberspace Security, Beijing University of Posts and Telecommunications. His research interests include in-network data processing, network security and information privacy issues in distributed systems, and the Internet of Things.

Jian Li received the Ph.D. degree from the Beijing Institute of Technology in 2005. He is currently a Professor with the School of Computer Science, Beijing University of Posts and Telecommunications. He has authored or coauthored 12 books and has published more than 100 professional research papers. His research interests include information security and quantum cryptography.

Bharat K. Bhargava (F'93) is currently a Professor of computer science at Purdue University. He has published hundreds of research papers and has received five best paper awards in addition to the Technical Achievement Award and Golden Core Award from the IEEE. He is conducting research in security and privacy issues in distributed systems and sensor networks. This involves identity management, trust and privacy, secure routing in Internet and mobile networks and dealing with malicious hosts, adaptability to attacks, controlled data dissemination, and experimental studies. He is the Editor-in-Chief of four journals and serves on over ten editorial boards of international journals. He is the Founder of the IEEE Symposium on Reliable and Distributed Systems, the IEEE Conference on Digital Library, and the ACM Conference on Information and Knowledge Management.