Analyzing and Evaluating Efficient Privacy-Preserving Localization for Pervasive Computing

Guanghui Wang, *Student Member, IEEE*, Jianping He, *Member, IEEE*, Xiufang Shi, *Member, IEEE*, Jianping Pan, *Senior Member, IEEE*, and Subin Shen, *Member, IEEE*

***Abstract*—Privacy-preserving localization in crowdsourcing has drawn much attention recently. Under the classical non-adjacent subtraction-based localization (NSL) model, existing solutions based on homomorphic encryption techniques are of high computational and communication overheads. In this paper, an adjacent subtraction-based localization (ASL) model is first proposed. Then, an efficient privacy-preserving localization (EPPL) algorithm is developed under ASL without using any homomorphic encryption technique. In terms of the correctness, privacy, and efficiency, a comprehensive analysis is presented to investigate EPPL's performance. Furthermore, the statistical equivalence between ASL and NSL is proved through the fact that the difference between their average location estimation results converges toward zero. The lower and upper bounds of the localization error are also derived for ASL under a bounded noise model. Extensive simulations are conducted to illustrate the equivalence between ASL and NSL, and the performance of EPPL regarding the correctness, privacy, and efficiency.**

***Index Terms*—Analysis and evaluation, crowdsourcing localization, pervasive computing, privacy preserving.**

I. Introduction

CROWDSOURCING is a new paradigm in pervasive computing era, which brings massive intelligence to solve difficult problems at an acceptable cost [1]–[3]. The central part in crowdsourcing is the combination of crowd and outsourcing, which aims to divide work among participants to achieve a cumulative solution with improved performance, including efficiency, scalability, and flexibility. Crowdsourcing localization provides a new way to capture the advantage of both powerful mobile devices and crowdsourcing in order to support ubiquitous localization. It applies the principles of crowdsourcing to perform the localization process involving multiple pervasive devices. The ranging-based crowdsourcing localization schemes in pervasive computing typically involve three phases: 1) anchor discovery; 2) distance ranging; and 3) location estimation [4]–[8]. First, the target node needs to have some anchor nodes to participate in the localization process. Second, the distances between the target and the anchors are measured. Third, the target node's location is estimated based on both measured distances and each anchor's location.

However, critical privacy concern is raised during a crowdsourcing localization process in pervasive computing. For example, the anchor users may be not willing to release their real location data to the target user due to privacy concern. The target user definitely does not want the anchor users to know the localization result. The location information is often a major privacy concern, since the anchor nodes and the target node in pervasive computing are just regular mobile users. Malicious users can correlate the user's location history and breach into many aspects of the user's private information, such as religious belief, health situation, hobby affiliation, daily agenda, and personal PIN [9]–[13]. Clearly, location information needs to be protected with caution. Without a reliable privacy protection scheme, many users would hesitate to participate in such a crowdsourcing localization process.

In recent years, many privacy-preserving localization (PPL) methods have been proposed in pervasive computing. For instance, based on homomorphic encryption techniques, the work in [14]–[18] designed various privacy-preserving methods for fingerprint-based localization. The work in [19] and [20] studied the privacy-preserving ranging-based localization problem. Specifically, the authors leveraged the combinations of information hiding and homomorphic encryption to develop PPL algorithms. However, directly applying homomorphic encryption techniques into crowdsourcing localization brings high computational and communication overheads.

Manuscript received July 1, 2017; revised October 3, 2017; accepted October 26, 2017. Date of publication November 10, 2017; date of current version August 9, 2018. This work was supported in part by the China Scholarship Council, in part by the Natural Science Foundation of China (NSFC) under Grant 61502246, in part by the College Graduate Innovation Projects of Jiangsu Province under Grant CXLX12_0487, in part by the Future Network Foresight Program of Jiangsu Province under Grant BY2013095-1-08, in part by the NUPTSF under Grant NY215019, in part by the Natural Sciences and Engineering Research Council of Canada, in part by the Canada Foundation for Innovation, and in part by the British Columbia Knowledge Development Fund. The work of J. He was supported by the NSFC under Grant 61773257. *(Corresponding author: Guanghui Wang.)*

G. Wang is with the College of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China, and also with the Department of Computer Science, University of Victoria, Victoria, BC V8P 5C2, Canada (e-mail: wanguvic@gmail.com).

J. He is with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: jphe@sjtu.edu.cn).

X. Shi is with the State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China (e-mail: xfshi.zju@gmail.com).

J. Pan is with the Department of Computer Science, University of Victoria, Victoria, BC V8P 5C2, Canada (e-mail: pan@uvic.ca).

S. Shen is with the School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: sbshen@njupt.edu.cn).

Digital Object Identifier 10.1109/JIOT.2017.2772291

This limitation of homomorphic encryption motivated us to design an efficient PPL (EPPL) algorithm that does not apply any homomorphic encryption technique. In this paper, an adjacent subtraction-based localization (ASL) model is first proposed after analyzing the classical nonadjacent subtraction-based localization (NSL) model. Then, in order to protect users' private information during the crowdsourcing localization, EPPL is developed with matrix decomposition and three random matrix approaches for summation, adjacent product summation, and adjacent difference summation. Furthermore, the performances of EPPL and ASL are thoroughly analyzed.

Specifically, this paper makes the following contributions.

1) An ASL model is proposed, whose adjacency property enables the suitability of efficiently protecting users' private information.
2) We prove that ASL and NSL are statistically equivalent by deriving that the difference of the two models' average localization results converges toward zero. We also derive the lower and upper bounds of the localization error for ASL by considering a bounded noise model.
3) Based on ASL, we develop an EPPL algorithm without using any homomorphic encryption technique. We also prove the correctness, privacy and efficiency of EPPL.

The rest of this paper is organized as follows. Section II provides the related work. In Section III, we present the system model ASL and formulate the problem. The EPPL algorithm is developed in Section IV. The performance analyses on ASL and EPPL are conducted in Sections V and VI, respectively. After evaluating the performance of ASL and EPPL via simulations in Section VII, we conclude this paper in Section VIII. This paper is extended from our earlier publication [21].

II. Related Work

A. Privacy-Preserving Localization

Existing PPL work can be classified into two types: 1) ranging-free PPL [14]–[18] and 2) ranging-based PPL [19], [20], [22]. Most existing ranging-free PPL work [14]–[16] focused on developing privacy-preserving WiFi or CSI fingerprint-based localization approaches with the homomorphic encryption techniques. Moreover, existing work [17], [18] applied a temporal vector map framework to enable PPL.

With regards to ranging-based PPL, Shu *et al.* [19], [20] considered multilateral PPL in pervasive computing and proposed three PPL protocols by leveraging the combination of homomorphic encryption and information hiding, which, however, is not an efficient way to achieve PPL due to high overheads. Furthermore, Hussain and Koushanfar [22] studied the PPL problem utilizing Yao's garbled circuit (GC) that allows two parties to jointly compute a function on their private inputs. Nevertheless, this GC-based approach was designed for smart automotive systems and is time consuming for pervasive computing with portable mobile devices.

Different from the above existing work, we analyze that the NSL-based PPL algorithms lead to high overheads due to the utilization of homomorphic encryption techniques. To solve this problem, we propose an ASL-based PPL algorithm which does not utilize any homomorphic encryption technique and hence reduces the computational and communication overheads, opening a new privacy-preserving paradigm for crowdsourcing localization in pervasive computing.

B. Localization Performance Analysis

The research on localization performance analysis mainly lies on the investigation of the effect of ranging noise and anchor distribution on localization accuracy.

The effect of ranging noise on localization error has been explored at two different levels: 1) experimental ranging noise distributions [23]–[25] and 2) theoretical relationships between ranging noise and localization error [26], [27]. For example, Wymeersch *et al.* [23] and Li *et al.* [24] applied experimental measurements to show that ranging noise did not exhibit a Gaussian distribution and focused on ranging error mitigation for localization. An unknown and bounded noise model was utilized to formulate a network localization problem in [25]. Regarding the theoretical relationships between ranging noise and localization error, Wei *et al.* [26] discussed the effect of ranging noise on localization error for a network localization problem. The localization accuracy under both additive and multiplicative noises was studied in [27].

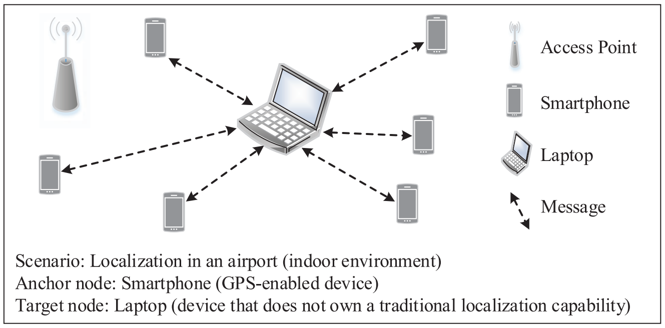
On the other hand, the localization error is also seriously affected by the anchor distribution [28]. Efforts have been made to analyze the optimal anchor distribution. For instance, Shames *et al.* [29] focused on reducing the effect of noisy measurement on formation localization. Bishop *et al.* [30] studied the optimal sensor-target localization geometries. Geometric forms of trilateration and outlier detection were considered for noisy localization in [6] and [31].

In this paper, we derive the deterministic expression for the ASL model. Through considering a bounded noise model, we obtain the localization lower and upper error bounds for ASL. Note that Cramer–Rao lower bound (CRLB) theory is a traditional method to analyze localization accuracy [32]. However, the CRLB theory is inappropriate in this paper since the bounded noise in this paper has no determined distribution.

III. System Model and Problem Formulation

A. Scenario and Localization Process

Scenario: The crowdsourcing localization in pervasive computing divides the localization work among participants in order to improve localization performance. Fig. 1 shows the scenario for the crowdsourcing localization with privacy concern in pervasive computing. In particular, multiple regular mobile anchor devices (e.g., GPS-enabled smartphones) which get access to the free WiFi in an airport environment, are involved in a localization process to help a target mobile device (e.g., laptop, sensor, or tablet that does not own a traditional localization capability) to calculate its location. However, these anchor devices' user-location information may be disclosed to the target device and the target device also has a concern on its location privacy. Therefore, it is necessary to protect the location privacy for both anchor devices



Scenario: Localization in an airport (indoor environment)
Anchor node: Smartphone (GPS-enabled device)
Target node: Laptop (device that does not own a traditional localization capability)

Fig. 1. Scenario for crowdsourcing localization in pervasive computing.

and target device simultaneously during the localization process.

The crowdsourcing localization consists of three phases: 1) anchor discovery; 2) distance ranging; and 3) location computation.

1) Anchor Discovery: Target node 0 recruits m mobile anchor nodes that help to calculate its location and denote them as nodes 1 to m, respectively. For node i ($i = 0, 1, \ldots, m$), its location is denoted by $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$, where n is the dimensionality of the space (normally $m \geq n+1$), and $\mathbf{x}_0$ needs to be calculated.

2) Distance Ranging: The distances between each anchor node and the target node are estimated. Two cases can be considered: 1) anchor ranging and 2) target ranging. For the former case, node i estimates the distance $d_{i,0}$ to node 0 and takes it as its private information. For the latter case, node 0 estimates the distance $d_{0,i}$ to node i and takes it as its private information. Since the privacy-preserving problem in anchor-ranging-based localization has been well studied in [19] and [20], we only consider target-ranging-based localization in this paper.

3) Location Computation: Based on the information of $(\mathbf{x}_i, d_{0,i})$, the multilateration method [33] is used to calculate the target node's location through minimizing the mean squared error (MMSE) between the ranging distances and the calculated coordinate-based distances.

In this paper, we aim to design a PPL algorithm with high efficiency during one crowdsourcing localization process. We do not consider multiple crowdsourcing localization processes where the target repeatedly recruits other users so that the target is able to estimate other users' locations. This case will be investigated in our future work. The important notations in this paper are listed in Table I.

B. Adjacent Subtraction-Based Localization

In order to securely calculate $\mathbf{x}_0$, the classical NSL model in [19], [20], and [33] used a nonadjacent subtraction-based approach to cancel the quadratic terms in the multilateration method. Specifically, based on the distance information, each node i ($i = 1, \ldots, m$) is supposed to satisfy a distance condition

$$\sqrt{\sum_{j=1}^{n} \left(x_{0,j} - x_{i,j}\right)^2} = d_{0,i} \tag{1}$$

TABLE I
IMPORTANT NOTATIONS

Notation	Description
$\mathbf{x}_i$	the location of node i, i.e., $\mathbf{x}_i = (x_{i,1}, x_{i,2}, ..., x_{i,n})$
$\widehat{\mathbf{x}}_0$	the MMSE estimate on the location of target node 0
m	the number of the anchor nodes
n	the dimensionality of the space
$d_{0,i}$	the ranging distance between node 0 and node i
$s_{0,i}$	the real distance between node 0 and node i
$\widehat{\epsilon}_A$	the localization error of ASL
δ_i	the measurement noise between node 0 to node i
δ_b	the measurement noise bound
$\mathbb{H}$, $\mathbb{Q}$	the coefficient and measurement matrices of ASL
$\mathbb{A}$, $\mathbb{B}$	the coefficient and measurement matrices of NSL
D	the determinant of $\mathbb{H}^T\mathbb{H}$
$r_{i,j}$	the coordinate difference of the j-th dimension of node i
g_i^l, g_i^u	the lower bound and upper bounds of g_i
e_i, f_i	the matrix decomposition terms $e_i = \sum_{j=1}^{n} x_{i,j}^2$, $f_i = d_{0,i}^2$

where $x_{0,j}$ ($j = 1, \ldots, n$) is the variable to be determined. To make the system easier for secure computation, the condition is rearranged as

$$\sum_{j=1}^{n} x_{0,j}^2 - 2\sum_{j=1}^{n} x_{0,j}x_{i,j} = d_{0,i}^2 - \sum_{j=1}^{n} x_{i,j}^2. \tag{2}$$

For m such conditions, the quadratic term in the left hand of the equation is canceled by subtracting the mth equation by the ith one. A linear system $\mathbb{A}\mathbf{x}_0^T = \mathbb{B}$ is obtained, where $\mathbb{A} = [A_1, \ldots, A_{m-1}]^T$, $\mathbb{B} = [b_1, \ldots, b_{m-1}]^T$

$$A_i = 2\left[x_{m,1} - x_{i,1} \quad \ldots \quad x_{m,n} - x_{i,n}\right] \tag{3}$$

$$b_i = \sum_{j=1}^{n}\left(x_{m,j}^2 - x_{i,j}^2\right) - \left(d_{0,m}^2 - d_{0,i}^2\right). \tag{4}$$

The MMSE estimate for $\mathbb{A}\mathbf{x}_0^T = \mathbb{B}$ is given by

$$\widehat{\mathbf{x}}_0^T = \left(\mathbb{A}^T\mathbb{A}\right)^{-1}\mathbb{A}^T\mathbb{B}. \tag{5}$$

Under the NSL model, each user needs to exchange information with the mth user when performing PPL. This nonadjacency property of NSL causes that NSL-based PPL has to apply homomorphic encryption techniques, e.g., Pailliar homomorphic encryption in [19] and [20], to protect private information.

The ASL model is introduced to design an EPPL algorithm without any homomorphic encryption technique. In particular, the quadratic term in (2) is canceled by subtracting the ith equation by the adjacent $(i+1)$th one. In this way, we can obtain a linear system $\mathbb{H}\mathbf{x}_0^T = \mathbb{Q}$, where $\mathbb{H} = [H_1, \ldots, H_{m-1}]^T$, $\mathbb{Q} = [q_1, \ldots, q_{m-1}]^T$

$$H_i = 2\left[x_{i,1} - x_{i+1,1} \ \ \ldots \ \ x_{i,n} - x_{i+1,n}\right] \tag{6}$$

$$q_i = \sum_{j=1}^{n}\left(x_{i,j}^2 - x_{i+1,j}^2\right) - \left(d_{0,i}^2 - d_{0,i+1}^2\right). \tag{7}$$

Based on the derived linear system $\mathbb{H}\mathbf{x}_0^T = \mathbb{Q}$, the MMSE estimate of the ASL model is given by

$$\widehat{\mathbf{x}}_0^T = \left(\mathbb{H}^T\mathbb{H}\right)^{-1}\mathbb{H}^T\mathbb{Q}. \tag{8}$$

The equivalence between the ASL and NSL models is proved in Section V-A. The difference between them is that they use adjacent and nonadjacent subtraction methods to cancel the quadratic terms, respectively. More importantly, the adjacency property of the ASL model enables us to design an EPPL algorithm without using any homomorphic encryption technique. However, the NSL-based PPL algorithm has high overheads due to the use of homomorphic encryption techniques.

C. Problem Formulation

In order to achieve PPL with high efficiency, the goal in this paper is to develop an EPPL algorithm based on the ASL model. Three performance metrics are considered: 1) correctness; 2) privacy preservation; and 3) efficiency.

1) *Correctness:* The target location calculated by the EPPL algorithm is equal to the MMSE estimate in (8).
2) *Privacy Preservation:* The EPPL algorithm is able to calculate target's location $\mathbf{x}_0$ without allowing node $j \neq i$, where $j = 0, 1, \ldots, m$ and $i = 0, 1, \ldots, m$, to learn the private information $\mathbf{x}_i$. Meanwhile, $d_{0,k}$ cannot be learned by node k, where $k = 1, \ldots, m$.
3) *Efficiency:* The EPPL algorithm has low computational and communication overheads comparing with existing homomorphic encryption technique-based PPL algorithms.

IV. EPPL Algorithm

A. Basic Idea

EPPL is an efficient privacy-preserving localization algorithm without using homomorphic encryption techniques. The key idea of EPPL is to decompose the computation in (8) through matrix decompositions. After the decompositions, (8) can be securely calculated by some random matrix schemes. In order to do the decompositions, (6) and (7) are rewritten as

$$H_i = 2(\mathbf{x}_i - \mathbf{x}_{i+1}) \tag{9}$$

$$q_i = (e_i - e_{i+1}) - (f_i - f_{i+1}) \tag{10}$$

where $e_i = \sum_{j=1}^n x_{i,j}^2$, and $f_i = d_{0,i}^2$. Note that e_i is node i's private information and f_i is node 0's private information ($i = 1, \ldots, m$). If we directly compute $\widehat{\mathbf{x}}_0$ using (8), it will leak the private information among nodes. Thus, based on the rewritten forms in (9) and (10), the intermediate terms $\mathbb{H}^T\mathbb{H}$ and $\mathbb{H}^T\mathbb{Q}$ can be decomposed as

$$\mathbb{H}^T\mathbb{H} = 4\left(\mathbf{x}_1^T\mathbf{x}_1 + 2\sum_{i=2}^{m-1}\mathbf{x}_i^T\mathbf{x}_i + \mathbf{x}_m^T\mathbf{x}_m\right) - 4\left(\sum_{i=1}^{m-1}\mathbf{x}_i^T\mathbf{x}_{i+1} + \sum_{i=1}^{m-1}\mathbf{x}_{i+1}^T\mathbf{x}_i\right) \tag{11}$$

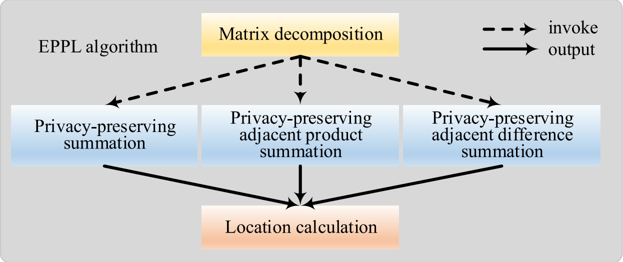

EPPL algorithm
Matrix decomposition
invoke
output
Privacy-preserving summation
Privacy-preserving adjacent product summation
Privacy-preserving adjacent difference summation
Location calculation

Fig. 2. EPPL algorithm structure.

$$\begin{aligned}\mathbb{H}^T\mathbb{Q} = {} & 2\left(e_1\mathbf{x}_1^T + 2\sum_{i=2}^{m-1}e_i\mathbf{x}_i^T + e_m\mathbf{x}_m^T\right) \\ & - 2\left(\sum_{i=1}^{m-1}e_i\mathbf{x}_{i+1}^T + \sum_{i=1}^{m-1}e_{i+1}\mathbf{x}_i^T\right) \\ & - 2\sum_{i=1}^{m-1}(f_i - f_{i+1})\left(\mathbf{x}_i^T - \mathbf{x}_{i+1}^T\right).\end{aligned} \tag{12}$$

An observation in the above equations shows that the first terms on the right-hand side (RHS) of both (11) and (12) need to securely calculate the *summation* of m nodes' private information. The second terms on the RHS of both equations need to securely calculate the *adjacent product summation* of m nodes' private information. Note that $\mathbf{x}_{i+1}^T\mathbf{x}_i$ is a matrix transposition of $\mathbf{x}_i^T\mathbf{x}_{i+1}$ and we just calculate them once. Furthermore, the third term on the RHS of (12) needs to securely calculate the *adjacent difference summation* of m nodes' private information. Therefore, in order to securely calculate $\widehat{\mathbf{x}}_0^T$, three privacy-preserving building blocks are needed for the above-mentioned summation, adjacent product summation, and adjacent difference summation operations.

B. EPPL Structure

Fig. 2 shows the structure of the EPPL algorithm. The privacy preservation is achieved through invoking three building blocks: 1) privacy-preserving summation (PPS); 2) privacy-preserving adjacent product summation (PPPS); and 3) privacy-preserving adjacent difference summation (PPDS). These building blocks are designed as follows.

1) PPS: Suppose node i has its private matrix M_i ($i = 1, \ldots, m$) and node 0 wants to calculate the summation $\alpha = \sum_{i=1}^m M_i$ without knowing any other nodes' private information. A random matrix scheme is used to secure the summation. In particular, each node i generates m random matrices p_i^k such that the summation of the matrices is zero, where p_i^k has the same size as M_i. One matrix is kept by node i and the rest matrices are sent to the rest nodes one by one. Thus, each node gets a constructed random matrix by adding these received matrices from other nodes to the kept matrix. The constructed random matrix of node i is denoted by P_i. Note that $\sum_{i=1}^m P_i = 0$. Then, node i sends the mixed information $\alpha_i = M_i + P_i$ to node 0. Without getting the private information, node 0 computes the summation of all private information by

$$\alpha = \sum_{i=1}^m \alpha_i. \tag{13}$$

2) PPPS: Suppose node i has its private vector $V_i = [v_1^i, \dots, v_n^i]$ $(i = 1, \dots, m)$ and node 0 wants to calculate the adjacent production summation $\beta = \sum_{i=1}^{m-1}(V_i^T V_{i+1})$ without knowing any other node's private information. In order to improve the efficiency, the product is converted to a summation through the logarithm function. In particular, a logarithm matrix L_i is constructed as

$$L_i = \begin{bmatrix} \log v_1^i & \dots & \log v_n^i \\ \vdots & \ddots & \vdots \\ \log v_1^i & \dots & \log v_n^i \end{bmatrix}. \tag{14}$$

All elements in V_i are positive (this can always hold since V_i denotes the coordinate of node i and can be shifted positive).

The random matrix scheme is also used to secure the adjacent product summation. First, each node i and its neighbor node $i+1$ construct their logarithm matrix L_i and L_{i+1} by (14), respectively. Second, node 0 generates one random matrix p_i with the same size as L_i and sends it to node i. Then, node i calculates and sends $U_i = p_i + L_i^T$ to node $i+1$. Node $i+1$ calculates and sends $U_{i+1} = U_i + L_{i+1}$ to node 0. Finally, node 0 retrieves the logarithm summation by subtracting p_i, and the product of elements through an exponentiation operation, on which the adjacent product $\beta_{i,i+1} = V_i^T V_{i+1}$ is calculated. Therefore, without knowing the private information, node 0 computes the adjacent product summation by

$$\beta = \sum_{i=1}^{m-1} \beta_{i,i+1}. \tag{15}$$

3) PPDS: Suppose node i has its private vector $V_i = [v_1^i, \dots, v_n^i]$ $(i = 1, \dots, m)$ and node 0 wants to calculate the adjacent difference summation $\gamma = \sum_{i=1}^{m-1}(V_i^T - V_{i+1}^T)$ without knowing any other nodes' private information. The random vector scheme is used to secure the adjacent difference summation in a similar way of the above PPPS, the key idea of which is an adjacent summation process. Hence, node i sends $U_i = V_i^T + p_i$ to node $i+1$, which sends $U_{i+1} = U_i - V_{i+1}^T$ to node 0. Then, node 0 retrieves $\gamma_{i,i+1} = V_i^T - V_{i+1}^T$ from the received information by subtracting p_i. Thus, without obtaining the private information, node 0 computes the adjacent difference summation by

$$\gamma = \sum_{i=1}^{m-1} \gamma_{i,i+1}. \tag{16}$$

C. EPPL Detail

The EPPL algorithm is described in Algorithm 1. Suppose that node i has its privacy concern on $\mathbf{x}_i$. Node 0 has its privacy concern on $\widehat{\mathbf{x}}_0$ and $d_{0,i}$. Node 0 wants to securely calculate $\widehat{\mathbf{x}}_0$. The input of Algorithm 1 is node i's location information $\mathbf{x}_i$, and the ranging information $d_{0,i}$. The output is the node 0's location which should be calculated without leaking the each node's private information. Based on (11) and (12), we make the following simplifications:

$$\Omega_1 = 4\mathbf{x}_1^T\mathbf{x}_1 + 8\sum_{i=2}^{m-1}\mathbf{x}_i^T\mathbf{x}_i + 4\mathbf{x}_m^T\mathbf{x}_m$$

Algorithm 1 EPPL

```
Input: $\mathbf{x}_i$, $d_{0,i}$, $i = 1, ..., m$;
Output: $\widehat{\mathbf{x}}_0$;
The $m$-th node calculates $\Omega_1$ and $\Omega_2$ by (13).
The $(m-1)$-th, $(m-2)$-th, and $(m-3)$-th nodes calculate $\psi_1$, $\psi_2$, and $\psi_3$ by (15), respectively.
Node 0 sends $f_{i,i+1} = f_i - f_{i+1}$ to the $(m-4)$-th node.
The $(m-4)$-th node calculates $\phi$ by (16).
Node 0 first calculates $\Theta = \Omega_1 - \psi_1 - \psi_1^T$ and $\Phi = \Omega_2 - \psi_2 - \psi_3 - \phi$ by (13). Then, it calculates $\widehat{\mathbf{x}}_0^T = \Theta^{-1}\Phi$.
```

$$\Omega_2 = 2e_1\mathbf{x}_1^T + 4\sum_{i=2}^{m-1} e_i\mathbf{x}_i^T + 2e_m\mathbf{x}_m^T$$

$$\psi_1 = 4\sum_{i=1}^{m-1}\mathbf{x}_i^T\mathbf{x}_{i+1},\ \psi_2 = 2\sum_{i=1}^{m-1} e_i\mathbf{x}_{i+1}^T,\ \psi_3 = 2\sum_{i=1}^{m-1} e_{i+1}\mathbf{x}_i^T$$

$$\phi = 2\sum_{i=1}^{m-1}(f_i - f_{i+1})\left(\mathbf{x}_i^T - \mathbf{x}_{i+1}^T\right).$$

Algorithm 1 utilizes five anchor nodes [let us say special anchor (S-Anchor) nodes] to achieve PPL based on the three privacy-preserving building blocks. The five nodes help node 0 to securely calculate $\widehat{\mathbf{x}}_0$ as follows. First, using PPS, one node securely calculates Ω_1 and Ω_2. Using PPPS, another three nodes securely calculate ψ_1, ψ_2, and ψ_3, respectively. The last node securely calculates ϕ using PPDS. Then, node 0 securely calculates two intermediate terms in (11) and (12) using PPS, and estimates its location using the MMSE method. The performance of Algorithm 1 is analyzed in Section VI.

V. Analysis of ASL: Equivalence and Accuracy

A. Equivalence Between ASL and NSL

Suppose that the distance measurement is the real distance plus a zero-mean measurement noise, which is given as $d_{0,i} = s_{0,i} + \delta_i$, where $s_{0,i}$ is the real distance, δ_i is the zero-mean measurement noise. Let $\widehat{\mathbf{x}}_A^T$ and $\widehat{\mathbf{x}}_N^T$ denote the MMSE estimation results under ASL and NSL, respectively. The statistical equivalence between ASL and NSL is proved through showing that the difference between the average of $\widehat{\mathbf{x}}_A$ and the average of $\widehat{\mathbf{x}}_N$ converges toward 0 with probability 1, which is given by the following theorem.

Theorem 1: Considering m anchor nodes and their ranging measurement $d_{0,i}$, $(i = 1, \dots, m)$, we have

$$\Pr\left\{\lim_{k\to\infty}\left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_A^{(t)} - \frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_N^{(t)}\right\| = 0\right\} = 1 \tag{17}$$

where k is the iteration number, and $\widehat{\mathbf{x}}_A^{(t)}$ and $\widehat{\mathbf{x}}_N^{(t)}$ are the estimation results at the tth iteration for ASL and NSL, respectively.

The proof of Theorem 1 can be found in Appendix A. According to the proof, one finds that both average of $\widehat{\mathbf{x}}_A$ and average of $\widehat{\mathbf{x}}_N$ converge toward $\mathbf{x}_0$ with probability 1. From Theorem 1, we have the following insights.

1) Error Equivalence Analysis: The localization errors for ASL and NSL are statistically equivalent. Specifically, using the location $\mathbf{x}_0 = (x_{0,1}, \ldots, x_{0,n})$ plus the localization error $\epsilon = (\epsilon_1, \ldots, \epsilon_n)$ to substitute $\widehat{\mathbf{x}}_A$ in the ASL model, a similar linear system $\mathbb{H}\epsilon^T = \mathbb{C}$ is derived where the ϵ is the unknown variable, $\mathbb{C} = [c_1, c_2, \ldots, c_{m-1}]^T$

$$c_i = q_i - 2\sum_{j=1}^{n}\left(x_{i,j} - x_{i+1,j}\right)x_{0,j}.$$

Then, using MMSE, ASL's localization error is obtained as

$$\widehat{\epsilon}_A^T = \left(\mathbb{H}^T\mathbb{H}\right)^{-1}\mathbb{H}^T\mathbb{C}. \tag{18}$$

Similarly, NSL's localization error can be obtained as

$$\widehat{\epsilon}_N^T = \left(\mathbb{A}^T\mathbb{A}\right)^{-1}\mathbb{A}^T\overline{\mathbb{C}} \tag{19}$$

where $\overline{\mathbb{C}} = [\overline{c}_1, \overline{c}_2, \ldots, \overline{c}_{m-1}]^T$

$$\overline{c}_i = b_i - 2\sum_{j=1}^{n}\left(x_{m,j} - x_{i,j}\right)x_{0,j}.$$

Using (18) and (19), it can be found that the localization errors are statistically equivalent based on the proof of Theorem 1.

2) Relationship Between Localization Error and Measurement Noise: The relationship between the localization error and the measurement noise is derived for ASL. In particular, through substituting $d_{0,i}$ with the real distance $s_{0,i}$ plus a measurement noise δ_i, and substituting the $\mathbf{x}_0$ with the real location $\mathbf{x}_0$ plus the localization error ϵ, the relationship between $\widehat{\epsilon}_A$ and δ_i under ASL is derived as

$$\widehat{\epsilon}_A^T = \left(\mathbb{H}^T\mathbb{H}\right)^{-1}\mathbb{H}^T\mathbb{C}' \tag{20}$$

where $\mathbb{C}' = [c'_1, c'_2, \ldots, c'_{m-1}]^T$

$$\begin{aligned} c'_i = &\sum_{j=1}^{n}\left(x_{i,j}^2 - x_{i+1,j}^2\right) - 2\sum_{j=1}^{n}\left(x_{i,j} - x_{i+1,j}\right)x_{0,j} \\ &- \left[\left(s_{0,i} + \delta_i\right)^2 - \left(s_{0,i+1} + \delta_{i+1}\right)^2\right]. \end{aligned} \tag{21}$$

From (20) and (21), one finds that both anchor location and measurement noise affect the localization accuracy.

B. Deterministic Expression of MMSE Estimate

Based on the MMSE estimate in (8), the deterministic expression of $\widehat{\mathbf{x}}_0^T$ is derived in a 2-D space.

Let $r_{i,j} = x_{i,j} - x_{i+1,j}$. According to the definition of $\mathbb{H}$, one gets that

$$\mathbb{H}^T\mathbb{H} = 4\begin{bmatrix} y_1 & y_3 \\ y_3 & y_2 \end{bmatrix}$$

where

$$y_1 = \sum_{i=1}^{m-1} r_{i,1}^2 \quad y_2 = \sum_{i=1}^{m-1} r_{i,2}^2 \quad y_3 = \sum_{i=1}^{m-1} r_{i,1}r_{i,2}.$$

Inverting $\mathbb{H}^T\mathbb{H}$, one gets that

$$\left(\mathbb{H}^T\mathbb{H}\right)^{-1} = \frac{4}{D}\begin{bmatrix} y_2 & -y_3 \\ -y_3 & y_1 \end{bmatrix}$$

where $D = 16[y_1y_2 - y_3^2]$ is the determinant of $\mathbb{H}^T\mathbb{H}$. Using (8), the expression of $\widehat{\mathbf{x}}_0^T$ is given as

$$\widehat{\mathbf{x}}_0^T = \frac{8}{D}\begin{bmatrix} y_2y_4 - y_3y_5 \\ y_1y_5 - y_3y_4 \end{bmatrix} \tag{22}$$

where

$$\begin{aligned} y_4 &= \sum_{i=1}^{m-1} r_{i,1}g_i, \; y_5 = \sum_{i=1}^{m-1} r_{i,2}g_i \\ g_i &= \left(x_{i,1}^2 + x_{i,2}^2 - x_{i+1,1}^2 - x_{i+1,2}^2\right) + \left(d_{0,i+1}^2 - d_{0,i}^2\right). \end{aligned}$$

Thus, the deterministic expression of $\widehat{\mathbf{x}}_0$ is obtained in (22), from which one finds the relationship between the location estimate and the distance measurements, and the relationship between the location estimate and the anchor nodes. It should be noted that the deterministic expression in (22) can be easily extended to a 3-D space.

The localization error $\widehat{\epsilon}_A^T$ for the ASL model is derived using (22). In particular, let $\widehat{\mathbf{x}}_0 = (\widehat{x}_{0,1}, \widehat{x}_{0,2})$ denote the estimated location of the target node and let $\mathbf{x}_0 = (x_{0,1}, x_{0,2})$ denote the location of the target node. The localization error vector of ASL is given as

$$\widehat{\epsilon}_A^T = \begin{bmatrix} \widehat{x}_{0,1} - x_{0,1} \\ \widehat{x}_{0,2} - x_{0,2} \end{bmatrix}. \tag{23}$$

The Euclidean norm $\|\widehat{\epsilon}_A\| = \sqrt{(\widehat{x}_{0,1} - x_{0,1})^2 + (\widehat{x}_{0,2} - x_{0,2})^2}$ is often taken as a scalar metric for localization accuracy.

C. Localization Accuracy

In order to analyze ASL's accuracy, similar to the work in [25], [34], and [35], a practical bounded noise model is used in the ranging measurement $d_{0,i} = s_{0,i} + \delta_i$, where δ_i satisfies

$$|\delta_i| \leq \delta_b \tag{24}$$

and δ_b is a positive bound.

The basic idea for deriving the localization error bounds is simple. First, with the bounded noise model and the deterministic expression in (22), the lower and upper bounds for $\{g_i, y_4, y_5\}$ are easy to obtain. Then, we get the lower and upper bounds for $\widehat{x}_{0,1}$ and $\widehat{x}_{0,2}$, which are used to bound the MMSE estimate. Finally, the localization error bounds can be obtained based on the bounds of $\widehat{x}_{0,1}$ and $\widehat{x}_{0,2}$. During the deriving process, we assume that all the anchor nodes are not collinear in the localization area.

According to (22), it should be noted that the lower and upper bounds will be affected by the sign of $r_{i,1}$, $r_{i,2}$, and y_3. Therefore, the Heaviside step function [36] is applied as

$$H(r) = \begin{cases} 0, & r < 0 \\ 1, & r \geq 0. \end{cases}$$

The lower and the upper bounds for g_i are given as

$$\begin{aligned} g_i^l &= \sum_{j=1}^{2}\left(x_{i,j}^2 - x_{i+1,j}^2\right) + \left(s_{0,i+1} + s_{0,i}\right)\left(s_{0,i+1} - s_{0,i} - 2\delta_b\right) \\ g_i^u &= \sum_{j=1}^{2}\left(x_{i,j}^2 - x_{i+1,j}^2\right) + \left(s_{0,i+1} + s_{0,i}\right)\left(s_{0,i+1} - s_{0,i} + 2\delta_b\right). \end{aligned}$$

The lower and upper bounds for y_4 are given as

$$y_4^l = \sum_{i=1}^{m-1}\Big[H\big(r_{i,1}\big)r_{i,1}g_i^l + H\big(-r_{i,1}\big)r_{i,1}g_i^u\Big]$$
$$y_4^u = \sum_{i=1}^{m-1}\Big[H\big(r_{i,1}\big)r_{i,1}g_i^u + H\big(-r_{i,1}\big)r_{i,1}g_i^l\Big].$$

The lower and upper bounds y_5^l, y_5^u can be derived similarly. Let $\{z_1^u, z_2^u\}$ denote the upper bounds for y_3y_5 and y_3y_4, where z_1^u is given as

$$z_1^u = H(y_3)y_3y_5^u + H(-y_3)y_3y_5^l$$

and z_2^u can be derived similarly. Thus, the lower bounds for $\widehat{x}_{0,1}$ and $\widehat{x}_{0,2}$ are given as

$$\widehat{x}_{0,1}^l = \frac{8}{D}\Big(y_2y_4^l - z_1^u\Big), \widehat{x}_{0,2}^l = \frac{8}{D}\Big(y_1y_5^l - z_2^u\Big).$$

With the lower bounds for $\widehat{x}_{0,1}$ and $\widehat{x}_{0,2}$, the following theorem is used to obtain the localization error bounds.

Theorem 2: Considering the ASL in a 2-D space and the ranging measurement noise satisfies (24), the localization error $\|\widehat{\epsilon}_A\|$ satisfies

$$0 \leq \|\widehat{\epsilon}_A\| < \sqrt{\Big(\widehat{x}_{0,1}^l - x_{0,1}\Big)^2 + \Big(\widehat{x}_{0,2}^l - x_{0,2}\Big)^2}. \tag{25}$$

Theorem 2 is proved in Appendix B. It provides the upper and lower bounds of the localization error for ASL. Based on its proof, one finds that the condition to reach the lower bound is that all the measurement noises are equal and all the distances between the anchor nodes and the target are equal. Even though the upper bound is not very tight, the term on the RHS of (25) is calculable if the noise bound δ_b and the anchor nodes $\mathbf{x}_i$ are determined. Therefore, (25) offers straightforward lower and upper bounds of the localization error.

VI. Analysis of EPPL: Correctness, Privacy, and Efficiency

A. Correctness Analysis

Theorem 3: Under the EPPL algorithm, the MMSE estimate $\widehat{\mathbf{x}}_0^T$ defined in $\mathbb{H}\mathbf{x}_0^T = \mathbb{Q}$ is correctly calculated.

The correctness of the EPPL algorithm means $\widehat{\mathbf{x}}_0^T$ is correctly calculated by the EPPL algorithm based on the ASL model in (8). Based on the matrix decomposition in (11) and (12) and the zero-sum property of the random matrices/vectors in the three building blocks, it is straightforward to obtain the above theorem.

B. Privacy Analysis

Some assumptions are made when analyzing the privacy for the EPPL algorithm. All the nodes in the localization process are honest but curious, where a node executes the computation specified by the algorithm, but is curious about whatever information of others that it could learn from the computation. The communication between two nodes is encrypted so that privacy leaking does not come from the communication process. Any active attack with a purpose of misleading or cheating the target is not considered.

There are two scenarios when performing the privacy analysis: 1) independent nodes and 2) colluding nodes. For the first scenario, a node can learn other nodes' privacy only based on the legitimate information that it receives. For the second scenario, colluding nodes are able to exchange their information to figure out more information about others.

Theorem 4: For independent nodes, the EPPL algorithm achieves PPL when $m > 6$.

Theorem 5: For colluding nodes, the EPPL algorithm achieves PPL if neither node 0 nor the five S-Anchor nodes involve in the collusion when $m > 6$.

Theorems 4 and 5 are proved in Appendixes C and D, respectively.

In the EPPL algorithm, to securely calculate (11) and (12), we use a few anchor nodes to participate into the calculation of the six terms $\{\Omega_1, \Omega_2, \psi_1, \psi_2, \psi_3, \phi\}$. The following theorem provides the minimum number of the S-Anchor nodes and the way to calculate the above six terms.

Theorem 6: The EPPL algorithm achieves PPL if and only if at least five S-Anchor nodes calculate $\{\Omega_1, \Omega_2\}, \{\psi_1\}, \{\psi_2\}, \{\psi_3\}$, and $\{\phi\}$, respectively, when $m > 6$.

Theorem 6 is proved in Appendix E. In the above three theorems, Theorems 4 and 5 show that the EPPL algorithm can achieve PPL in the independent and colluding scenarios, respectively. Theorem 6 studies the minimum number of the S-Anchor nodes and the way to use these nodes through providing a sufficient and necessary condition for the EPPL algorithm to achieve PPL.

In terms of privacy/performance tradeoff analysis, the EPPL algorithm is compared with the PPL algorithm in [19] and [20] which is based on the homomorphic encryption technique. For independent nodes, both EPPL algorithm and PPL algorithm achieve PPL. The condition for the EPPL algorithm is that it requires $m > 6$, while the condition for the PPL algorithm is that it requires $m > n$. For colluding nodes, both algorithms achieve PPL under different conditions. The EEPL algorithm requires that the target node and the five S-Anchor nodes do not involve in the colluding, while the PPL algorithm requires that the number of colluding anchor nodes is fewer than half of $(m - 1)$ and the number of noncolluding anchor nodes is greater than $(n + 1)$. Thus, EPPL achieves PPL and does not use any homomorphic encryption technique.

C. Efficiency Analysis

The computational and communication overheads of the EPPL algorithm are analyzed. The efficiency analysis follows the methodology in [19] and [20], where the computational overhead is dominated by the vector operations and the communication overhead is dominated by the transmission of elements.

1) Computational Overhead: For the anchor nodes that do not participate in the computation [let us say ordinary anchor (O-Anchor) nodes], their computational overheads are dominated by vector multiplication and logarithm operations when performing the building block operations. The total computational overhead on an O-Anchor node is $3n$ logarithm operations and n^2+n multiplication operations. Then, for the S-Anchor nodes, their computational overheads are dominated by not only the vector multiplication and logarithm operations in the building blocks, but also the exponentiation operations in (14). Thus, the computational overhead on an S-Anchor node is at most $3n$ logarithm operations, n^2+n multiplication operations, and $(m-1)n^2$ exponentiation operations.

The computational overhead on the target node is examined through analyzing vector multiplication operations and matrix inversion operations. In particular, the target node needs to perform one matrix inversion operation (n^4 multiplication), and n^3 multiplications in step 5 of Algorithm 1. Therefore, the computational overhead is n^4+n^3 multiplications.

2) Communication Overhead: The communication overhead is dominated by the transmission of elements. In step 1 of Algorithm 1, calculating Ω_1 needs to transmit $(m-1)(m-2)n^2+(m-1)n^2$ elements. The number of transmitted elements for calculating Ω_2 can be similarly obtained as $(m-1)(m-2)n+(m-1)n$. In step 2 of Algorithm 1, calculating ψ_1 needs to transmit $3(m-1)n^2$ elements. The number of transmitted elements for calculating ψ_2, ψ_3 can be similarly obtained as $3(m-1)n$ and $3(m-1)n$. In addition, steps 3–5 involve $(m-1)$, $3(m-1)n$, and $4n^2+7n$ elements, respectively. Thus, the total number of elements is roughly $(n^2+n)m^2+(n^2+7n)m+2n^2$ per localization process.

VII. Performance Evaluation

A. Simulation Setup

In order to verify the results, extensive simulations are conducted using MATLAB software which runs on a DELL desktop PC with Intel core 2 Quad CPU Q9450 @ 2.66-GHz processor and 4.00 GB (3.24 GB available) RAM. We consider a 500×500 m^2 square area. The target node is located at the center of the square and the anchor nodes are randomly distributed in the area under a uniform distribution. The ranging distance is simulated as the real distance plus a ranging noise $d_{0,i} = s_{0,i} + \delta_i$. For the verification of the equivalence between ASL and NSL, the zero-mean noise δ_i is assumed as a Gaussian distribution with mean $\mu = 0$ and standard deviation $\sigma = 5$ m. For the verification of the localization accuracy, the bounded noise δ_i is assumed as a uniform distribution in $[-\delta_b, \delta_b]$, where $\delta_b = \{0.5, 1.0, 1.5, 2.0\}$ m. For each simulation, we perform 1000 independent runs and report the average of the estimation results due to the randomness of the random variable. The localization error is calculated by $\|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|$, where $\bar{\mathbf{x}}_0$ is the average estimated location.

B. Equivalence Evaluation Between ASL and NSL

First, the statistical equivalence between ASL and NSL is shown through the localization error and running time with the increasing number of anchors. Fig. 3 illustrates the localization error and running time. The running time is an average of the time over 1000 independent runs in order to show the running time of a single simulation. In Fig. 3(a), it can be found that most of the localization errors (less than 0.2 m) of both models are close enough because of the statistical equivalence between them. Furthermore, in Fig. 3(b), it can be seen that both localization models have a similar running time also because of their statistical equivalence.

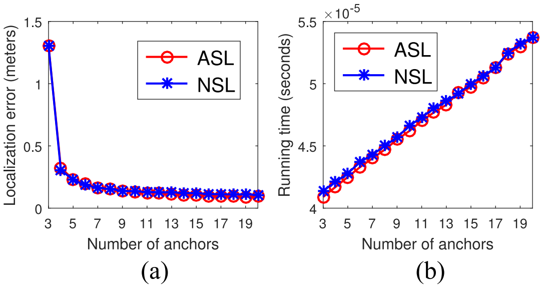


Fig. 3. Equivalence evaluation with the increment of anchors. (a) Localization errors. (b) Running time of each run.

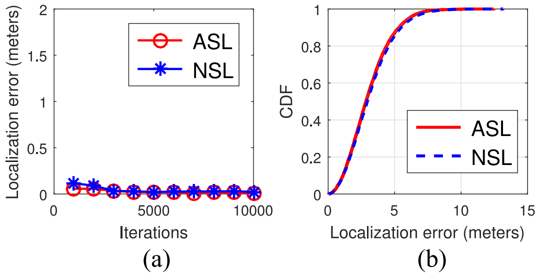


Fig. 4. Equivalence evaluation on localization errors. (a) Average localization errors with iterations. (b) CDFs.

Then, the average localization errors with iterations and their cumulative distribution functions (CDFs) are plotted for ASL and NSL in Fig. 4. The largest iteration number is 10 000. Based on the average results in Fig. 4(a), the average errors of both ASL and NSL models decrease with the increment of iterations. When the iteration number reaches 3000, there is a marginal difference between the ASL and NSL models. Fig. 4(b) presents the CDFs of ASL and NSL. It is observed that the distribution of ASL is close enough to that of NSL. Specifically, almost 85% of the localization errors are less than 5 m for both ASL and NSL. Both results verify that the two models are statistically equivalent.

The above simulations demonstrate the statistical equivalence between ASL and NSL, which validates Theorem 1 in Section V-A. Thus, based on ASL, our EPPL algorithm has an equivalent localization performance with the existing NSL-based PPL algorithm.

C. Localization Accuracy Evaluation

First, the localization error bounds are demonstrated by the bounded localization area under the bounded noise. Ten random anchors are used to calculate the bounded localization area with $\delta_b = \{0.5, 1.0, 1.5, 2.0\}$ m. Fig. 5 shows four bounded areas by simulations. It is obvious that when the noise bound increases, the bounded area increases. Furthermore, the bounded area is a rectangle where the target node is located at the center, which confirms our viewpoint in the proof of Theorem 2. In particular, when $\delta_b = 2.0$ m, the rectangle area *ABCD* is the largest area. The length of the side *AB* is about 10 m, where the coordinate of the *x*-axis for the target is at

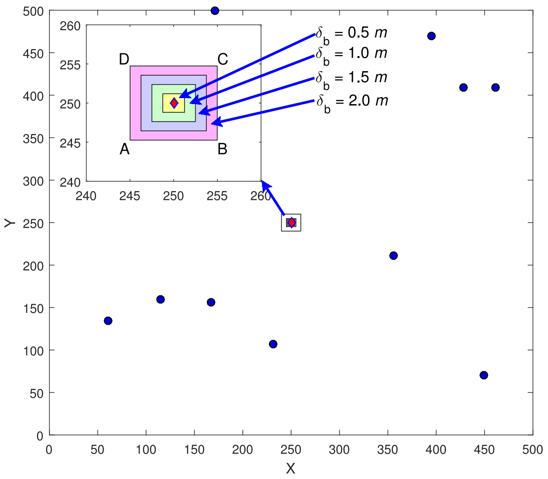


Fig. 5. Localization area bound with bounded noises and ten random anchors.

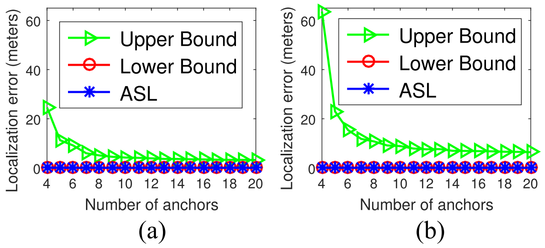


Fig. 6. Localization error bounds with anchors and bounded noises. (a) $\delta_b = 1$ m. (b) $\delta_b = 2$ m.

the middle point of the side *AB*. The coordinate of the *y*-axis for the target is at the middle point of the side *AD*.

Then, we study the relationship between the number of anchors and the localization error bounds. The number of anchors is from 4 to 20 and the noise bound $\delta_b = \{1, 2\}$ m. Fig. 6 illustrates the localization error bounds with the increment of the anchors. Fig. 6(a) presents the upper and lower bounds and also the localization error for ASL when $\delta_b = 1$ m. The upper and lower bounds are calculated using Theorem 2. It is observed that the localization error is definitely below the upper bound and the lower bound is below the localization error. In particular, the localization error is closer to the lower bound than the upper bound, which verifies that the upper bound in (25) is not very tight. Fig. 6(b) presents the similar results when $\delta_b = 2$ m. It is found that when the noise bound is doubled, the upper bounds are also almost doubled because the upper bound in (25) is directly related to the noise bound when the anchor nodes are fixed. Furthermore, according to both Fig. 6(a) and (b), it is seen that the upper bound decreases when the number of anchors increases. This is because that more anchors provide more information about the target when the noise bound is fixed. Therefore, these results validate the theoretical work in Theorem 2.

Finally, we compare the localization performance with noise bound $\delta_b = \{1, 2\}$ m in Fig. 7. The CDFs of the localization errors are plotted with 1000 independent simulations. The average errors are plotted with the increasing number of iterations. The simulations use 20 random anchors. In Fig. 7(a), it is noted that the results with $\delta_b = 1$ m have a better performance than that with $\delta_b = 2$ m. In particular, almost 90% of the localization errors when $\delta_b = 1$ m are less than 0.5 m, while only 40% of the localization errors are less than this value when $\delta_b = 2$ m. This is because with the fixed number of anchors and the fixed locations, the noise bound is the only factor that affects the localization results, which is easily observed from Theorem 2. Fig. 7(b) plots the average localization errors with iterations and different noise bounds. It is found that with the increment of the iterations, the average localization errors for both $\delta_b = 1$ m and $\delta_b = 2$ m are extremely similar. This is because the both bounded noises are uniformly distributed in $[-\delta_b, \delta_b]$, of which the mean is zero. Thus, even though the noise bounds are different, a large number of iterations are able to mitigate the influence of the bounded noise on the average localization error. This also validates that the MMSE estimate of the ASL model converges with probability 1 toward the real location of the target, which has been proved in Theorem 1.

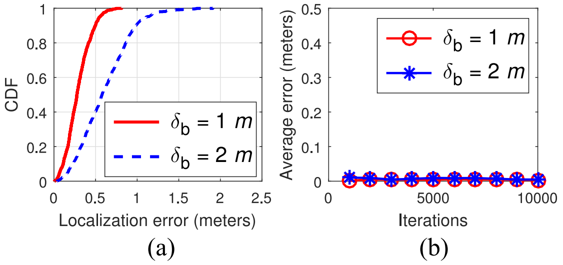


Fig. 7. Localization errors with bounded noises. (a) Localization error CDFs. (b) Average errors.

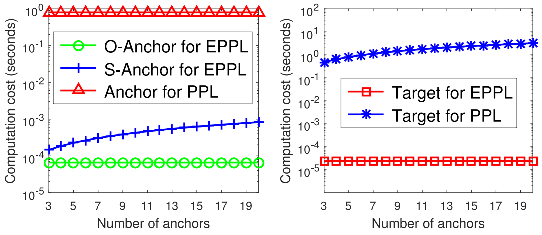
Based on the above simulations, one sees that the performance of the proposed ASL model is shown with bounded noise. The localization area bound is obtained, the relationship between the localization error and the noise bound is evaluated, and the average localization error converges toward zero with probability 1. Therefore, the proposed ASL model can guarantee the correctness and accuracy of the proposed EPPL algorithm which is based on the ASL model.

D. Numerical Results for EPPL

Table II presents the comparisons of the computational and communication overheads between the EPPL algorithm and the PPL algorithm in [19] and [20]. Following the setting in [19] and [20], we assume that a real number is represented in 24 bits. The notations of χ_1, χ_2, ε_1, ε_2, and ε_3 represent the operations of 24-bit multiplication, 2048-bit multiplication, 24-bit exponentiation or logarithm, 1024-bit exponentiation, and 2048-bit exponentiation, respectively. In our numerical results, we assume the following execution time for these operations: $\chi_1 = 1$ μs, $\chi_2 = 0.88$ ms, $\varepsilon_1 = 10$ μs, $\varepsilon_2 = 81.08$ ms, and $\varepsilon_3 = 159.06$ ms. The setting of these parameters also follows [19] and [20], which are obtained from real experiments in [37]. We assume the communication between nodes has a bandwidth of 2 Mb/s. Table II also compares the execution time of EPPL and PPL, which are analyzed based on the time consumed in each step, including both computational and communication overheads. The computational and communication overheads in steps 1, 2, and 4 of the EPPL algorithm

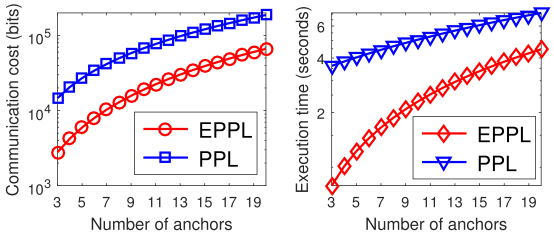
TABLE II
COMPARISONS ON COMPUTATIONAL AND COMMUNICATION OVERHEADS AND EXECUTION TIME

Algorithm	EPPL	PPL
Computation on ordinary anchor	$3n\varepsilon_1 + (n^2 + n)\chi_1$	$2\varepsilon_2 + 2n\varepsilon_3 + n\chi_2 + (3n^2 + 4n)\chi_1$
Computation on special anchor	$[3n + (m-1)n^2]\varepsilon_1 + (n^2 + n)\chi_1$	
Computation on target	$(n^4 + n^3)\chi_1$	$m\chi_2 + m\varepsilon_3 + (n^4 + n^3)\chi_1$
Communication	$[(n^2 + n)m^2 + (n^2 + 7n)m + 2n^2] \times 24$	$2048mn + [m^2n^2 + (3n+1)m^2] \times 24$
Execution time	$[3n + (m-1)n^2]\varepsilon_1 + (n^4 + n^3 + n^2 + n)\chi_1$ $+ (4n^2m + mn + 6n) \times 24/2000$	$(n^4 + n^3 + 4n^2 + 5n)\chi_1 + (n+1)\chi_2 + (2n+1)\varepsilon_3$ $+ 2\varepsilon_2 + [2048n + 24 \times (2n^2m + 4mn + m)]/2000$



(a) (b)

Fig. 8. Efficiency evaluation on computational cost. (a) Cost of the anchors. (b) Cost of the target.



(a) (b)

Fig. 9. Efficiency evaluation on communication cost and execution time. (a) Communication cost. (b) Execution time.

are counted once by calculating ψ_1, since the five S-Anchor nodes are able to execute in parallel, and calculating ψ_1 incurs the most overhead.

We plot the anchor node's computational cost as a function of the number of anchor nodes in Fig. 8(a). In particular, the computational cost on both S-Anchor node and O-Anchor node in the EPPL algorithm is much lower than that in the PPL algorithm. This is because the homomorphic encryption-based PPL algorithm needs to perform encryption/decryption by long-bit multiplication and exponentiation operations. The computational costs on each anchor of PPL and each O-Anchor of EPPL change little with the number of anchors because the computational costs of these nodes are independent of the number of anchor nodes as shown in Table II. Hence, each anchor node in the EPPL algorithm has small computational cost, which indicates that the EPPL algorithm is able to attract more users to participate in the localization process in order to improve the accuracy of the localization result. Furthermore, it can be found that the computational cost of the S-Anchor node is a bit higher than that of the O-Anchor node and increases with the number of anchors, where the reason is that the computational cost of the S-Anchor calculating ψ_1 is dependent on the number of anchors. In Fig. 8(b), the target node's computational cost is plotted. The target node in the EPPL algorithm is much more computationally efficient (at least four orders of magnitude) than that in the PPL algorithm. Specifically, the computational cost of the target node in the EPPL algorithm changes little with the number of anchors because computations are distributed into the five S-Anchor nodes. The PPL algorithm has much computational cost because it is based on the homomorphic encryption technique.

The communication cost and the execution time of EPPL and PPL are compared in Fig. 9. In particular, it can be seen that in Fig. 9(a) the communication cost of the EPPL algorithm is smaller than that of the PPL algorithm because its calculation does not need the homomorphic encryption process. In Fig. 9(b), the execution time of the EPPL algorithm is much smaller than that of the PPL algorithm because of no homomorphic encryption process. Specifically, when the numbers of anchors are 3 and 20, the EPPL algorithm reduces the execution time by 78.1% and 37.0%, respectively. With the number of anchors from 3 to 20, the EPPL algorithm reduces the execution time by 53.3% on average.

The above numerical results illustrate the efficiency of the EPPL algorithm. In terms of computational cost, communication cost, and execution time, EPPL always has a better performance than PPL. It should be noted that EPPL can work in a 3-D space since we consider n-dimensional space when both designing EPPL and analyzing the correctness, privacy, and efficiency of EPPL.

E. Location Privacy Preservation Evaluation

The location privacy is measured by the probability that the colluding anchors are able to learn other nodes' location information. The location privacy preservation strength is defined as the probability that the colluding anchors cannot calculate other nodes' location information, which can be calculated based on combinatorial probability analyses. First, there are two kinds of anchor nodes in the EPPL algorithm: 1) five S-Anchor nodes and 2) other O-Anchor nodes. Then, according to the privacy analyses in Section VI-B, the colluding anchors cannot learn other nodes' locations if the S-Anchors do not involve in the collusion. This means that the privacy preservation strength can be calculated by the probability of only the O-Anchor nodes involving in the collusion. Under the definition of the location privacy strength, it can be seen that the larger is the probability, the stronger is the location privacy. Using this definition, we evaluate the target and anchor location privacy preservation strength for the proposed EPPL algorithm and the the existing PPL algorithm in [19] and [20].

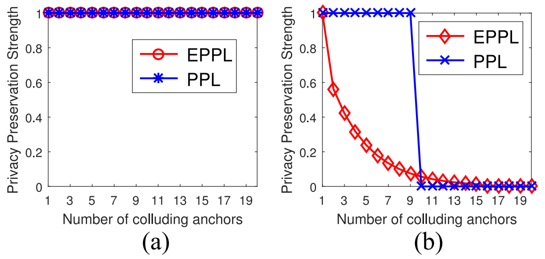


Fig. 10. Location privacy preservation strength. (a) Target location privacy. (b) Anchor location privacy

The location privacy preservation evaluations use 20 anchor nodes. In the target location privacy evaluation, we assume that an anchor node will try to obtain the estimation of the target location by colluding with some randomly selected anchor nodes. In the anchor location privacy evaluation, we assume that an anchor (or target) node wants to calculate another anchor node's location by colluding with some selected anchor (or target) nodes. During the evaluation, the location privacy preservation strength is measured when varying the number of the colluding anchor nodes.

The evaluation results on the target and anchor location privacy preservation strength are plotted in Fig. 10. For the target location privacy in Fig. 10(a), it is found that both EPPL and PPL algorithms have the same maximum privacy preservation strength, because the colluding anchor nodes in both algorithms do not have the distance ranging information and they cannot estimate the target location. For the anchor location privacy in Fig. 10(b), it can be seen that the privacy preservation strength of EPPL is degraded with the increase of the colluding anchor nodes. Let the number of the colluding anchor nodes be C. When $\mathsf{C} = 1$, the colluding anchor node cannot learn other anchor nodes' locations and the location privacy preservation strength is 1. When $2 \leq \mathsf{C} \leq m - 5$, the probability of one S-Anchor node being selected to participate in a collusion is $p = (5/m)$, which depends on the numbers of S-Anchors and total anchor nodes. The only way to preserve anchor location information is that all colluding anchor nodes are O-Anchor nodes, whose probability is $(1 - p)^{\mathsf{C}}$. When $\mathsf{C} > m - 5$, the S-Anchor nodes are going to involve in the collusion and the anchor privacy preservation strength becomes 0. Furthermore, in Fig. 10(b), it is also noted that when the number of the colluding anchor nodes is larger than 9, the anchor location privacy preservation strength of the PPL algorithm becomes 0 because the colluding anchor nodes can construct sufficient linear equations to obtain the locations of the noncolluding anchor nodes.

Based on the above evaluations, one finds that the EPPL algorithm has a comparable privacy preservation strength with the existing PPL algorithm in terms of the target location privacy. The anchor privacy preservation strength of EPPL is related to the total number of anchor nodes. Using more anchor nodes to participate in the EPPL algorithm, we can obtain stronger privacy preservation strength. The anchor location privacy preservation problem in the face of colluding nodes will be considered in our future work.

VIII. Conclusion

In this paper, to enable an EPPL in crowdsourcing, an ASL model is proposed to replace the NSL model. An EPPL algorithm is proposed without using the time-consuming homomorphic encryption techniques. Then, we prove that ASL and NSL are statistically equivalent through the fact that the difference between their average localization results converges toward zero with probability 1. The lower and upper bounds of ASL's localization error are analyzed by considering a bounded noise model. Furthermore, based on the correctness, privacy, and efficiency analyses, the EPPL algorithm is shown to correctly estimate the target's location and efficiently preserve the private information. Moreover, extensive simulations are conducted to demonstrate the proved equivalence and the localization accuracy. Numerical results show that the EPPL algorithm reduces the localization execution time by 53.3% on average. Location privacy preservation evaluations illustrate that EPPL achieves a comparable privacy preservation strength in terms of the target location privacy.

Appendix A
Proof of Theorem 1

The idea of the proof is explained as follows. First we prove that averages of both $\widehat{\mathbf{x}}_A$ and $\widehat{\mathbf{x}}_N$ converge toward $\mathbf{x}_0$ with probability 1. Then, (17) can be proved through a triangle inequality.

First, we focus on proving the following equations:

$$\Pr\left\{\lim_{k\to\infty}\left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_A^{(t)} - \mathbf{x}_0\right\| = 0\right\} = 1 \tag{26}$$

$$\Pr\left\{\lim_{k\to\infty}\left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_N^{(t)} - \mathbf{x}_0\right\| = 0\right\} = 1. \tag{27}$$

Due to the zero-mean noise δ_i and the definition of $\mathbb{Q}$, using the strong law of large numbers, we have that

$$\Pr\left\{\lim_{k\to\infty}\left|\frac{1}{k}\sum_{t=1}^{k} d_{0,i}^{(t)} - s_{0,i}\right| = 0\right\} = 1$$

$$\Pr\left\{\lim_{k\to\infty}\left|\underbrace{\frac{1}{k}\sum_{t=1}^{k} q_i^{(t)}}_{t_1} - \underbrace{\left[\sum_{j=1}^{n}\left(x_{i,j}^2 - x_{i+1,j}^2\right) - \left(s_{0,i}^2 - s_{0,i+1}^2\right)\right]}_{t_2}\right| = 0\right\} = 1.$$

From the above equation, term t_1 converges with probability 1 toward to term t_2. Based on the ASL model $\widehat{\mathbf{x}}_A^T = (\mathbb{H}^T\mathbb{H})^{-1}\mathbb{H}^T\mathbb{Q}$, we get that the average of $\widehat{\mathbf{x}}_A$ converges to $\mathbf{x}_0$ with probability 1. Thus, (26) is proved. In a similar way, (27) is also proved.

Second, based on the following inequality:

$$\left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_A^{(t)} - \frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_N^{(t)}\right\| \leq \left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_A^{(t)} - \mathbf{x}_0\right\| + \left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_N^{(t)} - \mathbf{x}_0\right\|$$

one finds that

$$\begin{aligned}&\lim_{k\to\infty}\left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_A^{(t)}-\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_N^{(t)}\right\|\\&\quad\le\lim_{k\to\infty}\left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_A^{(t)}-\mathbf{x}_0\right\|+\lim_{k\to\infty}\left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_N^{(t)}-\mathbf{x}_0\right\|.\end{aligned}\tag{28}$$

Suppose that event E_1 is

$$\lim_{k\to\infty}\left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_A^{(t)}-\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_N^{(t)}\right\|=0$$

and event E_2 is

$$\lim_{k\to\infty}\left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_A^{(t)}-\mathbf{x}_0\right\|=0\text{ and }\lim_{k\to\infty}\left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_N^{(t)}-\mathbf{x}_0\right\|=0.$$

Based on (28), it is known that if E_1 is true, E_2 is not always true. If E_2 is true, E_1 is always true, which means that the probability of E_1's happening is no less than the probability of E_2's happening. Thus, we have

$$\begin{aligned}&\Pr\left\{\lim_{k\to\infty}\left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_A^{(t)}-\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_N^{(t)}\right\|=0\right\}\\&\ge\Pr\left\{\lim_{k\to\infty}\left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_A^{(t)}-\mathbf{x}_0\right\|=0;\lim_{k\to\infty}\left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_N^{(t)}-\mathbf{x}_0\right\|=0\right\}\\&=\Pr\left\{\lim_{k\to\infty}\left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_A^{(t)}-\mathbf{x}_0\right\|=0\right\}\\&\quad\times\Pr\left\{\lim_{k\to\infty}\left\|\frac{1}{k}\sum_{t=1}^{k}\widehat{\mathbf{x}}_N^{(t)}-\mathbf{x}_0\right\|=0\right\}\end{aligned}$$

which is equal to 1 according to (26) and (27). Therefore, Theorem 1 is proved. ■

Appendix B
Proof of Theorem 2

The proof is to prove that the distance between the $(\widehat{x}_{0,1},\widehat{x}_{0,2})$ and $(x_{0,1},x_{0,2})$ is smaller than the distance between $(\widehat{x}_{0,1}^l,\widehat{x}_{0,2}^l)$ and $(x_{0,1},x_{0,2})$. First we need to derive the lower and upper bounds of $\widehat{x}_{0,1}$ and $\widehat{x}_{0,2}$. Then, we prove that $(x_{0,1},x_{0,2})$ is at the center of the rectangle $\{\widehat{x}_{0,1}^l\le\widehat{x}_{0,1}\le\widehat{x}_{0,1}^u,\widehat{x}_{0,2}^l\le\widehat{x}_{0,2}\le\widehat{x}_{0,2}^u\}$, where $\widehat{x}_{0,1}^u$ and $\widehat{x}_{0,2}^u$ are the upper bounds for $\widehat{x}_{0,1}$ and $\widehat{x}_{0,2}$, respectively. Finally, we analyze the bounded conditions for the localization error to reach the bounds.

First, using (22), one gets that

$$\widehat{x}_{0,1}=\frac{8}{D}(y_2y_4-y_3y_5),\widehat{x}_{0,2}=\frac{8}{D}(y_1y_5-y_3y_4).$$

Since all the anchor nodes are not collinear, we have that $D\neq0$. Based on the Cauchy–Schwarz inequality, we have that $D>0$. Furthermore, we have $y_1>0$ and $y_2>0$. However, according to (22), the positive or negative $r_{i,1},r_{i,2},y_3$ affect the localization error bound derivation. For example, when $r_{i,1}>0$, the lower bound for y_4 is $r_{i,1}g_i^l$. When $r_{i,1}<0$, the lower bound for y_4 is $r_{i,1}g_i^u$. Thus, in order to obtain the lower and upper bounds for $\widehat{x}_{0,1}$ and $\widehat{x}_{0,2}$, we should be careful with the positive or negative $r_{i,1},r_{i,2}$, and y_3. Using the Heaviside step function $H(r)$, the lower bounds $\{z_1^l,z_2^l\}$ can be obtained in a similar way as obtaining the upper bounds in (25). Thus, the lower and upper bounds of g_i are needed to derive $\{z_1^l,z_2^l,z_1^u,z_2^u,y_4^l,y_5^l,y_4^u,y_5^u\}$.

Based on the bounded noise model in (24) and the definition of g_i in (22), we get that

$$\begin{aligned}\left(s_{0,i}-\delta_b\right)^2&\le d_{0,i}^2\le\left(s_{0,i}+\delta_b\right)^2\\\left(s_{0,i+1}-\delta_b\right)^2&\le d_{0,i+1}^2\le\left(s_{0,i+1}+\delta_b\right)^2\\g_i&\ge\sum_{j=1}^{2}\left(x_{i,j}^2-x_{i+1,j}^2\right)\\&\quad+\left(s_{0,i+1}+s_{0,i}\right)\left(s_{0,i+1}-s_{0,i}-2\delta_b\right)\\g_i&\le\sum_{j=1}^{2}\left(x_{i,j}^2-x_{i+1,j}^2\right)\\&\quad+\left(s_{0,i+1}+s_{0,i}\right)\left(s_{0,i+1}-s_{0,i}+2\delta_b\right).\end{aligned}$$

Thus, we obtain the lower bound and the upper bound for g_i, based on which the lower bounds and the upper bounds for y_4 and y_5 can be derived, followed by the lower bounds and the upper bounds for $\widehat{x}_{0,1}$ and $\widehat{x}_{0,2}$. In particular, the lower bounds for $\widehat{x}_{0,1}$ and $\widehat{x}_{0,2}$ are defined as those in (25) and the upper bounds are given as

$$\widehat{x}_{0,1}^u=\frac{8}{D}\left(y_2y_4^u-z_1^l\right),\widehat{x}_{0,2}^u=\frac{8}{D}\left(y_1y_5^u-z_2^l\right).$$

With the derived lower and upper bounds, the rectangle is obtained as $\{\widehat{x}_{0,1}^l\le\widehat{x}_{0,1}\le\widehat{x}_{0,1}^u,\widehat{x}_{0,2}^l\le\widehat{x}_{0,2}\le\widehat{x}_{0,2}^u\}$. For example, suppose that a rectangle area $ABCD$ denotes the bounded area of $(\widehat{x}_{0,1},\widehat{x}_{0,2})$. Let $(\widehat{x}_{0,1}^l,\widehat{x}_{0,2}^l)$, $(\widehat{x}_{0,1}^u,\widehat{x}_{0,2}^l)$, $(\widehat{x}_{0,1}^u,\widehat{x}_{0,2}^u)$, and $(\widehat{x}_{0,1}^l,\widehat{x}_{0,2}^u)$ denote the coordinates of vertex A,B,C, and D. Therefore, if $(x_{0,1},x_{0,2})$ is at the center of the rectangle $ABCD$, we get that the distance between $(\widehat{x}_{0,1}^l,\widehat{x}_{0,2}^l)$ and $(x_{0,1},x_{0,2})$ is not smaller than the distance between $(\widehat{x}_{0,1},\widehat{x}_{0,2})$ and $(x_{0,1},x_{0,2})$.

Next, in order to prove that $(x_{0,1},x_{0,2})$ is at the center of the rectangle area, we show that $\{\widehat{x}_{0,1}^l+\widehat{x}_{0,1}^u=2x_{0,1};\widehat{x}_{0,2}^l+\widehat{x}_{0,2}^u=2x_{0,2}\}$. Specifically, with the derived expression of the lower and upper bounds, we have that

$$\begin{aligned}\widehat{x}_{0,1}^l+\widehat{x}_{0,1}^u&=\frac{8}{D}\left[y_2\left(y_4^l+y_4^u\right)-\left(z_1^u+z_1^l\right)\right]\\z_1^u+z_1^l&=H(y_3)y_3\left(y_5^l+y_5^u\right)+H(-y_3)y_3\left(y_5^l+y_5^u\right)\\y_4^l+y_4^u&=\sum_{i=1}^{m-1}\left[H\left(r_{i,1}\right)r_{i,1}\left(g_i^l+g_i^u\right)+H\left(-r_{i,1}\right)r_{i,1}\left(g_i^l+g_i^u\right)\right]\\y_5^l+y_5^u&=\sum_{i=1}^{m-1}\left[H\left(r_{i,2}\right)r_{i,2}\left(g_i^l+g_i^u\right)+H\left(-r_{i,2}\right)r_{i,2}\left(g_i^l+g_i^u\right)\right]\\g_i^l+g_i^u&=2g\left[\sum_{j=1}^{2}\left(x_{i,j}^2-x_{i+1,j}^2\right)+\left(s_{0,i+1}+s_{0,i}\right)\left(s_{0,i+1}-s_{0,i}\right)\right].\end{aligned}$$

From the above equations and (22), one finds that the measurement noise δ_b is canceled and $g_i^l+g_i^u=2g_i$. In a similar way, $\widehat{x}_{0,2}^l+\widehat{x}_{0,2}^u$ can be obtained. Then, there is no measurement noise in the ASL model and the real location $(x_{0,1},x_{0,2})$

is determined by $\widehat{x}^l_{0,1}+\widehat{x}^u_{0,1}$ and $\widehat{x}^l_{0,2}+\widehat{x}^u_{0,2}$. Thus, we get that $\widehat{x}^l_{0,1}+\widehat{x}^u_{0,1}=2x_{0,1}$. In a similar way, we get $\widehat{x}^l_{0,2}+\widehat{x}^u_{0,2}=2x_{0,2}$. Thus, $(x_{0,1},x_{0,2})$ is at the center of the rectangle $ABCD$, which proves that the distance between $(\widehat{x}^l_{0,1},\widehat{x}^l_{0,2})$ and $(x_{0,1},x_{0,2})$ is not smaller than the distance between the $(\widehat{x}_{0,1},\widehat{x}_{0,2})$ and $(x_{0,1},x_{0,2})$. By now, we have proved that localization error is bounded in (25).

Finally, we analyze the bounded conditions for the localization error. For the lower bound, the condition for $\|\widehat{\epsilon}_A\|=0$ is that $\{\delta_i=\delta_{i+1}, s_{0,i}=s_{0,i+1}, i=1,\ldots,m-1\}$, which means that all the measurement noises are equal and all the distances between the anchor nodes and the target are equal. Note that this does not mean that if all the measurements are equal, it is able to derive $\|\widehat{\epsilon}_A\|=0$. For the upper bound, it is necessary to point out that $\|\widehat{\epsilon}_A\|$ cannot reach its upper bound. Suppose that $(\widehat{x}_{0,1},\widehat{x}_{0,2})$ is able to reach $(\widehat{x}^l_{0,1},\widehat{x}^l_{0,2})$, which means that y_4 reaches its lower bound and y_3y_5 reaches its upper bound. If $y_3>0$, it needs y_5 reaches its upper bound. Note that y_4 and y_5 cannot reach their lower and upper bounds $\{y^l_4, y^u_5\}$ simultaneously when $y_3>0$ which means that there exists at least one $k\in\{1,2,\ldots,m-1\}$ to make $r_{k,1}r_{k,2}>0$. Suppose that $\{r_{k,1}>0, r_{k,2}>0\}$, which means that in order to reach $\{y^l_4, y^u_5\}$, g_k needs to reach both g^l_k for y^l_4 and g^u_k for y^u_5. This is contradictory since $(\widehat{x}_{0,1},\widehat{x}_{0,2})$ cannot reach $(\widehat{x}^l_{0,1},\widehat{x}^l_{0,2})$. It is easy to verify that other cases are also contradictory, which proves that $\|\widehat{\epsilon}_A\|$ cannot reach the upper bound.

Therefore, Theorem 2 is proved. ■

APPENDIX C
PROOF OF THEOREM 4

The proof is to show that: 1) no anchor can learn the location of another anchor; 2) no anchor can learn the location of the target, not even obtaining a rough estimation about the location; and 3) the target cannot learn the location of any anchor.

Argument 1) contains two cases: 1) O-Anchor nodes and 2) the five anchor nodes that participate in the computation. The first case is obvious because the only information exchange between any two O-Anchors i and j, where $1\le i,j\le m-6$, is the random matrices in (13) or mixed information by random matrices/vectors in (15) and (16).

For the second case, we only show that the mth, $(m-1)$th, and $(m-4)$th nodes cannot calculate other nodes' locations since the proof for other nodes can be similarly obtained. The only way that the mth node can calculate other nodes' locations is to solve a linear equation set constructed from the received information α^1_i,α^2_i, and the calculated information Ω_1 and Ω_2. By treating $\mathbf{x}_i$ $(i\neq m)$, P^1_i, P^2_i as variables, the total number of the variables is $(m-1)n+(m-1)n^2+(m-1)n$. The total number of independent linear equations the mth anchor node may obtain is at most $(m-1)n^2+(m-1)n+n^2+n$, which is obtained by knowing α^1_i,α^2_i, and the relationship $\sum_{i=1}^{m-1}P^1_i=0$, $\sum_{i=1}^{m-1}P^2_i=0$. It can be seen that the number of variables is greater than the number of equations when $m>6$. Therefore, the mth anchor node cannot calculate the location of any other anchor node. For the $(m-1)$th anchor node calculating ψ_1, it is able to obtain more equations since $\mathbf{x}^T_i\mathbf{x}_{i+1}$ is an $n\times n$ matrix with n^2 equations while there are only $2n$ variables in $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. For example, when calculating $\mathbf{x}^T_1\mathbf{x}_2$, 4 equations can be obtained with variables $\{x_{1,1},x_{1,2},x_{2,1},x_{2,2}\}$ from the calculated matrix which is given by

$$\begin{bmatrix} x_{1,1}x_{2,1} & x_{1,1}x_{2,2} \\ x_{1,2}x_{2,1} & x_{1,2}x_{2,2} \end{bmatrix}. \tag{29}$$

It is impossible to calculate the variables from the equations because of the dependence between the equations. Thus, the $(m-1)$th anchor node also cannot calculate other nodes' locations. For the $(m-4)$th anchor node calculating ϕ, it can obtain $(m-2)n+(m-1)$ equations from the received $\mathbf{x}_i-\mathbf{x}_{i+1}$ and $f_{i,i+1}$. The variable number is $(m-1)n+m$ based on $\mathbf{x}_i$ and $d_{0,i}$. It is easy to find out the $(m-4)$th anchor node also cannot calculate other nodes' locations.

Argument 2) is true. From the matrix decomposition, it is clear that calculating the estimation $\widehat{\mathbf{x}}^T_0$, requires Ω_1, Ω_2, ψ_1, ψ_2, ψ_3, and ϕ. For independent nodes, no single anchor node has all such information. Especially, since all the ranging information is the target node's private information, it is impossible for any anchor node to guess $\widehat{\mathbf{x}}^T_0$.

Argument 3) can be proved as follows. First, the target node only learns $\Theta=\Omega_1-\psi_1-\psi^T_1$ and $\Phi=\Omega_2-\psi_2-\psi_3-\phi$ but cannot learn each term since Θ and Φ are calculated by (13). Then, with Θ, Φ, and the ranging distance $d_{0,i}$, the target obtains n^2+n+m equations and the variable number is mn from $\mathbf{x}_i$. It can be found that the variable number is larger than the equation number when $m>6$.

Combining the above arguments, Theorem 4 is proved. ■

APPENDIX D
PROOF OF THEOREM 5

Since both target node and five S-Anchor nodes do not involve in any collusion, the proof is to show that: 1) the colluding O-Anchor nodes cannot obtain a rough estimation about $\widehat{\mathbf{x}}^T_0$ and 2) the colluding O-Anchor nodes cannot calculate any other anchor nodes' locations.

Argument 1) is correct because in the EPPL algorithm, the target takes ranging distances as its private information, and each anchor only has its own location information. The colluding nodes cannot construct a small-scale multilateration linear system to estimate the target's location.

Argument 2) can be proved by noting that the information exchange between two anchors is random matrices in (13) or mixed information in (15) and (16), which is sent to the noncolluding special nodes. Thus, the colluding nodes cannot calculate any other nodes' locations.

Combining the above arguments, Theorem 5 is proved. ■

APPENDIX E
PROOF OF THEOREM 6

The proof is to show: 1) at least five anchor nodes ⇒ PPL and 2) PPL ⇒ at least five anchor nodes.

Argument 1) is obvious since Theorems 4 and 5 use five anchor nodes to achieve PPL.

Argument 2) is proved by showing that $\{\Omega_1, \Omega_2\}$, $\{\psi_1\}$, $\{\psi_2\}$, $\{\psi_3\}$, and $\{\phi\}$ must be separately calculated by at least five anchor nodes. According to the three privacy-preserving building blocks in (13), (15), and (16), we show the following two cases: 1) $\{\psi_1, \psi_2\}$ or $\{\psi_1, \phi\}$ or 2) $\{\psi_1, \Omega_1\}$ or $\{\phi, \Omega_1\}$ cannot calculated by one anchor node, and the target cannot participate in calculating any term. For the first case, if one node calculates $\{\psi_1, \psi_2\}$, it is possible that the node calculates the locations of other nodes. When calculating ψ_1, we use (29) as an example. Based on the obtained equations, one of the four variables $\{x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}\}$ can be used to express other three variables. When calculating ψ_2, we get

$$\begin{bmatrix} e_i x_{2,1} \\ e_i x_{2,2} \end{bmatrix}. \tag{30}$$

It is easy to find that the variables can be calculated from (29) and (30). Thus, $\{\psi_1, \psi_2\}$ cannot be calculated by one anchor node. In a similar way, it can be proved that other combinations in this case also cannot calculated by one anchor node. For the second case, if the target participates in calculating one term (let us say Ω_1), based on argument 3) in the proof of Theorem 4, the target can obtain $2n^2 + n + m$ equations and the variable number is mn. It is possible that the target learns other nodes' locations by solving the equations when $m > 6$. Thus, the target cannot participate in calculating any term. Combining the above two cases, argument 2) is correct.

Combining the above arguments, Theorem 6 is proved. ■

References

[1] K. Yang, K. Zhang, J. Ren, and X. Shen, "Security and privacy in mobile crowdsourcing networks: Challenges and opportunities," *IEEE Commun. Mag.*, vol. 53, no. 8, pp. 75–81, Aug. 2015.

[2] A. Tamilin, I. Carreras, E. Ssebaggala, A. Opira, and N. Conci, "Context–aware mobile crowdsourcing," in *Proc. ACM UbiComp*, 2012, pp. 717–720.

[3] I. Constandache, X. Bao, M. Azizyan, and R. Choudhury, "Did you see Bob? Human localization using mobile phones," in *Proc. ACM MobiCom*, 2010, pp. 149–160.

[4] H. Liu *et al.*, "Push the limit of WiFi based localization for smartphones," in *Proc. ACM Mobicom*, 2012, pp. 305–316.

[5] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," *Wireless Netw.*, vol. 10, no. 6, pp. 701–710, 2004.

[6] Z. Yang and Y. Liu, "Quality of trilateration: Confidence-based iterative localization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 5, pp. 631–640, May 2010.

[7] I. Guvenc and C.-C. Chong, "A survey on TOA based wireless localization and NLOS mitigation techniques," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 3, pp. 107–124, 3rd Quart., 2009.

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

[9] K. G. Shin, X. Ju, Z. Chen, and X. Hu, "Privacy protection for users of location-based services," *IEEE Wireless Commun.*, vol. 19, no. 1, pp. 30–39, Feb. 2012.

[10] C. Wang, X. Guo, Y. Wang, Y. Chen, and B. Liu, "Friend or Foe? Your wearable devices reveal your personal PIN," in *Proc. ACM ASIA CCS*, 2016, pp. 189–200.

[11] Y. Wang, Y. Chen, F. Ye, J. Yang, and H. Liu, "Towards understanding the advertiser's perspective of smartphone user privacy," in *Proc. IEEE ICDCS*, Columbus, OH, USA, 2015, pp. 288–297.

[12] H. Li, Y. He, X. Cheng, H. Zhu, and L. Sun, "Security and privacy in localization for underwater sensor networks," *IEEE Commun. Mag.*, vol. 53, no. 11, pp. 56–62, Nov. 2015.

[13] F. Buccafurri, G. Lax, S. Nicolazzo, and A. Nocera, "A privacy-preserving localization service for assisted living facilities," *IEEE Trans. Services Comput.*, to be published.

[14] H. Li, L. Sun, H. Zhu, X. Lu, and X. Cheng, "Achieving privacy preservation in WiFi fingerprint-based localization," in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, 2014, pp. 2337–2345.

[15] X. Wang, Y. Liu, Z. Shi, X. Lu, and L. Sun, "A privacy-preserving fuzzy localization scheme with CSI fingerprint," in *Proc. IEEE GLOBECOM*, San Diego, CA, USA, 2015, pp. 1–6.

[16] P. Armengol, R. Tobkes, K. Akkaya, B. Çiftler, and I. Güvenç, "Efficient privacy-preserving fingerprint-based Indoor localization using crowdsourcing," in *Proc. IEEE MASS*, Dallas, TX, USA, 2015, pp. 549–554.

[17] A. Konstantinidis *et al.*, "Privacy-preserving indoor localization on smartphones," in *Proc. IEEE ICDE*, Helsinki, Finland, 2016, pp. 1470–1471.

[18] A. Konstantinidis *et al.*, "Privacy-preserving indoor localization on smartphones," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 3042–3055, Nov. 2015.

[19] T. Shu, Y. Chen, J. Yang, and A. Williams, "Multi-lateral privacy-preserving localization in pervasive environments," in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, 2014, pp. 2319–2327.

[20] T. Shu, Y. Chen, and J. Yang, "Protecting multi-lateral localization privacy in pervasive environments," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1688–1701, Oct. 2015.

[21] G. Wang, J. Pan, J. He, and S. Shen, "An efficient privacy-preserving localization algorithm for pervasive computing," in *Proc. IEEE ICCCN*, Vancouver, BC, Canada, 2017, pp. 1–9.

[22] S. U. Hussain and F. Koushanfar, "Privacy preserving localization for smart automotive systems," in *Proc. ACM/EDAC/IEEE DAC*, Austin, TX, USA, 2016, pp. 1–6.

[23] H. Wymeersch, S. Marano, W. M. Gifford, and M. Z. Win, "A machine learning approach to ranging error mitigation for UWB localization," *IEEE Trans. Commun.*, vol. 60, no. 6, pp. 1719–1728, Jun. 2012.

[24] S. Li, M. Hedley, and I. B. Collings, "New efficient indoor cooperative localization algorithm with empirical ranging error model," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1407–1417, Jul. 2015.

[25] X. Shi, G. Mao, B. D. O. Anderson, Z. Yang, and J. Chen, "Robust localization using range measurements with unknown and bounded errors," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 4065–4078, Jun. 2017.

[26] M. Wei, R. Aragues, C. Sagues, and G. C. Calafiore, "Noisy range network localization based on distributed multidimensional scaling," *IEEE Sensors J.*, vol. 15, no. 3, pp. 1872–1883, Mar. 2015.

[27] X. Shi, R. Zhang, and Z. Yang, "Localization accuracy of range-only sensors with additive and multiplicative noise," *Math. Problems Eng.*, vol. 2014, Mar. 2014, Art. no. 956895.

[28] Y. Chen, J. Yang, W. Trappe, and R. Martin, "Impact of anchor placement and anchor selection on localization accuracy," in *Handbook of Position Location: Theory, Practice, and Advances*, S. A. Zekavat and M. Buehrer, Eds. Hoboken, NJ, USA: Wiley, 2011.

[29] I. Shames, B. Fidan, and B. D. O. Anderson, "Minimization of the effect of noisy measurements on localization of multi-agent autonomous formations," *Automatica*, vol. 45, no. 4, pp. 1058–1065, 2009.

[30] A. N. Bishop, B. Fidan, B. D. O. Anderson, K. Doğançay, and P. N. Pathirana, "Optimality analysis of sensor-target localization geometries," *Automatica*, vol. 46, no. 3, pp. 479–492, 2010.

[31] Z. Yang, L. Jian, C. Wu, and Y. Liu, "Beyond triangle inequality: Sifting noisy and outlier distance measurements for localization," *ACM Trans. Sensor Netw.*, vol. 9, no. 2, pp. 1–20, Apr. 2013.

[32] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, vol. 1. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.

[33] A. Savvides, C. Han, and M. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proc. ACM MobiCom*, 2001, pp. 166–179.

[34] J. He, X. Duan, P. Cheng, L. Shi, and L. Cai, "Accurate clock synchronization in wireless sensor networks with bounded noise," *Automatica*, vol. 81, pp. 350–358, Jul. 2017.

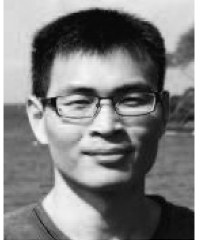
[35] J. He, M. Zhou, P. Cheng, L. Shi, and J. Chen, "Discrete consensus under bounded noise: An algorithm with fast convergence and high accuracy," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 2874–2884, Dec. 2016.

[36] E. Weisstein. *Heaviside Step Function, MathWorld A Wolfram Web Resource*. Accessed: Jun. 5, 2017. [Online]. Available: http://mathworld.wolfram.com/HeavisideStepFunction.html

[37] R. Zhang, J. Zhang, Y. Zhang, J. Sun, and G. Yan, "Privacy-preserving profile matching for proximity-based mobile social networking," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 656–668, Sep. 2013.

Guanghui Wang (S'17) received the bachelor's degree in mathematics and applied mathematics from the School of Mathematics and Information Science, North China University of Water Resources and Electric Power, Zhengzhou, China, in 2011. He is currently pursuing the Ph.D. degree in information networks at the Nanjing University of Posts and Telecommunications, Nanjing, China.

Since 2016, he has been a Visiting Research Student with the Department of Computer Science, University of Victoria, Victoria, BC, Canada. His current research interests include privacy-preserving localization, crowd sourcing, pervasive computing, and Internet of Things.



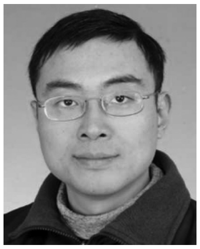
Jianping He (M'15) received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2013.

He is currently an Associate Professor with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. His current research interests include control and optimization of sensor networks and cyber-physical systems, scheduling and optimization in VANETs and social networks, and investment decision in financial market and electricity market.

Dr. He served as an Associate Editor for a Special Issue on Consensus-Based Applications in Networked Systems in the *International Journal of Robust and Nonlinear Control* in 2016.

Xiufang Shi (M'17) received the B.Sc. degree in automation from the East China University of Science and Technology, Shanghai, China, in 2011, and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2016. She was a joint Ph.D. student at the University of Sydney, Sydney, NSW, Australia, in 2015.

She is currently a Post-Doctoral Researcher with the College of Control Science and Engineering, Zhejiang University. Her current research interests include wireless localization, target tracking, wireless sensor network, and statistical signal processing.



Jianping Pan (S'96–M'98–SM'08) is currently a Professor of computer science with the University of Victoria, Victoria, BC, Canada. He was a Post-Doctoral Researcher with the University of Waterloo, Waterloo, ON, Canada. He was also with Fujitsu Laboratories of America, Sunnyvale, CA, USA, and NTT Laboratories. His current research interests include computer networks, distributed systems, protocols for advanced networking, performance analysis of networked systems, and applied network security.

Mr. Pan was a recipient of the IEICE Best Paper Award in 2009, the Telecommunications Advancement Foundation's Telesys Award in 2010, the JSPS Invitation Fellowship in 2012, and the Best Paper Award for WCSP'11, IEEE Globecom'11, and IEEE ICC'13. He has been serving on the Technical Program Committee of major computer communications and networking conferences, including IEEE INFOCOM, ICC, Globecom, WCNC, and CCNC. He is the Ad Hoc and Sensor Networking Symposium Co-Chair of IEEE Globecom'12 and an Associate Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He is a Senior Member of the ACM.

Subin Shen (S'95–A'97–M'02) received the bachelor's, master's, and Ph.D. degrees in computer science and engineering from Southeast University, Nanjing, China.

He is a Professor with the School of Computer and the School of Software, Nanjing University of Posts and Telecommunications, Nanjing. His current research interests include computer networks, telecommunication networks, the Internet of Things, cloud computing, big data, and future networks.

Dr. Shen is an Editor of Recommendation Y.2066 "Common Requirements of Internet of Things," Recommendation Y.2068 "Functional Framework and Capabilities of the Internet of Things," and Recommendation Y.2078 "Application Support Models of the Internet of Things" of the International Telecommunication Union, Telecommunication Standardization Sector. He is a member of the ACM, the China Computer Federation, and the China Institute of Communications.