An Efficient Privacy-Preserving Localization Algorithm for Pervasive Computing

Guanghui Wang*†, Jianping Pan†, Jianping He‡ and Subin Shen*
*College of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China
†Department of Computer Science, University of Victoria, BC, Canada
‡Department of Automation, Shanghai Jiao Tong University, Shanghai, China
Email: wanguvic@gmail.com, pan@uvic.ca, jphe@sjtu.edu.cn, sbshen@njupt.edu.cn

***Abstract*—Protecting location privacy of mobile systems is important for various location-based services in pervasive computing scenarios. How to quickly compute the target user's location without knowing each anchor user's location has drawn much attention. Under the classical nonadjacent subtraction based localization model, existing solutions based on homomophic encryption introduce much computation and communication overheads to achieve privacy-preserving localization. In this paper, an adjacent subtraction based localization model is proposed, which is suitable to efficiently protect users' privacy. Then, under such a model, an efficient privacy-preserving localization algorithm is developed without using homomophic encryption. A closed-form expression of the relationship between the localization error and the measurement noise is derived. Furthermore, a comprehensive analysis, including correctness analysis, privacy analysis, and efficiency analysis, is presented. Some simulations are conducted to show that the proposed model has equivalent accuracy and efficiency with the classical model. Some numerical results are presented to show the efficiency of the proposed privacy-preserving localization algorithm.**

I. Introduction

With the development of mobile systems in pervasive computing, mobile smart devices (e.g., smartphones and tablets) equipped with a variety of sensors have entered people's lives. Mobile smart devices are pervasive nowadays, and have the ability to provide a rich range of Location-Based Services (LBSs) to users [1], [2]. To support these applications, mobile smart devices first determine their locations for various tasks. In particular, a mobile user needs to know his/her location when performing Points Of Interest (POI) searching tasks in order to access the merchandise and promotion information.

The ranging-based localization schemes in pervasive computing typically involve three phases: anchor discovery, ranging and location estimation [3]–[5]. First, the target node needs to have some anchor nodes to participate in the localization process. Second, distances between the target and the anchors are measured. Then, the target node's location is estimated based on both the measured distances and each anchor's location. However, the anchor users may be not willing to release their real location data to the target user due to privacy concern. The location information is often major privacy concern, since the anchor nodes in pervasive computing are just regular mobile nodes (e.g., GPS-enabled smartphones). Malicious users can correlate user's location with the POIs that the users visited and breach into many aspects of the user's personal life, such as religious belief, health situation, hobby affiliation, daily agenda, and personal PIN [6]–[10]. Clearly, location information needs to be protected with caution. Without a reliable privacy protection scheme, many anchor users would hesitate to participate in such a localization process.

In recent years, many privacy-preserving localization methods have been proposed for pervasive computing [11]–[18]. In particular, the authors in [11] first analyzed the privacy issues of WiFi fingerprint-based localization and proposed a privacy-preserving WiFi fingerprint localization scheme. Then, the authors in [12] combined homomorphic encryption with a fuzzy logic-based approach to develop a privacy-preserving scheme for Channel State Information (CSI) based fingerprinted localization. The authors in [13] implemented a fingerprint-based localization scheme that utilized homomorphic cryptography to ensure the privacy of the client. Furthermore, a k-Anonymity Bloom (kAB) filter and a Temporal Vector Map (TVM) algorithm were used to solve the privacy-preserving problem for indoor WiFi fingerprint localization in [14], [15]. In addition, Siam U. Hussain et al. [16] focused on solving a privacy-preserving (triangle) localization problem through Yao's Garbled Circuit (GC) protocol. However, most of those privacy-preserving methods are designed for the fingerprint-based localization and cannot be applied into the ranging-based localization which is also a practical scenario.

The work in [17], [18] first studied the privacy-preserving ranging-based localization problem in pervasive environments. Specifically, the authors leveraged the combinations of information hiding and homomorphic encryption to develop privacy-preserving localization algorithms. However, applying homomophic encryption technique into pervasive localization is not an efficient way to achieve privacy-preserving localization due to high overheads.

In this work, we propose an Efficient Privacy-Preserving Localization (EPPL) algorithm without applying homomorphic encryption technique. In particular, an Adjacent Subtraction based Localization (ASL) model is proposed after analyzing the classical Nonadjacent Subtraction based Localization (NSL) model (see Theorem 1 for the equivalence proof). Under the ASL model, the EPPL algorithm is designed using three privacy-preserving algorithms of summation, adjacent product summation, and adjacent difference summation. Furthermore, a closed-form expression of the relationship between

978-1-5090-2991-4/17/$31.00 ©2017 IEEE

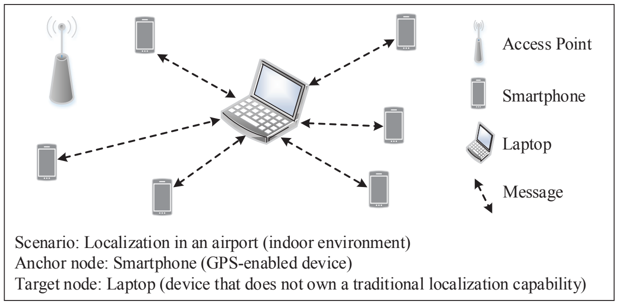
Access Point
Smartphone
Laptop
Message
Scenario: Localization in an airport (indoor environment)
Anchor node: Smartphone (GPS-enabled device)
Target node: Laptop (device that does not own a traditional localization capability)

Figure 1. Scenario for privacy-preserving localization

the localization error and the measurement noise is derived based on the equivalence analysis between the ASL model and the NSL model. In summary, the contributions of this paper are as follows:

- An adjacent subtraction based localization model ASL is proposed, which is more suitable to efficiently protect users' privacy than the classical nonadjacent subtraction based localization model NSL. We prove that the ASL and NSL models are equivalent.
- An efficient privacy-preserving localization algorithm EPPL is developed without using homomorphic encryption. We provide correctness, privacy, and efficiency analysis for the EPPL algorithm.
- A closed-form expression of the relationship between the localization error and the measurement noise is derived.

The rest of this paper is organized as follows. In Section II, we present some preliminaries and formulate the problem. Then, the EPPL algorithm is developed and analyzed in Section III and Section IV, respectively. After evaluating the performance of the EPPL algorithms in Section V, we conclude the paper in Section VI.

II. Preliminaries and Problem Formulation

In this section, we provide some important preliminaries and formulate the problem.

A. Scenario and Localization Process

Scenario: Figure 1 shows the scenario for privacy-preserving localization in pervasive computing. In particular, multiple regular mobile anchor devices (e.g., GPS-enabled smartphones) which access 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 do 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 has a concern on its location privacy. Therefore, it is necessary to protect the privacy for both the anchor devices and the target device simultaneously during the localization process.

The important notations in this paper are listed in Table I.

Table I
Important Notations

Notation	Description
$\mathbf{x}_i$	Node i's location, $\mathbf{x}_i = (x_{i,1}, x_{i,2}, ..., x_{i,n})$
$\widehat{\mathbf{x}}_0$	MMSE estimation on target node 0's location
$d_{0,i}$	Ranging distance between target node 0 and anchor node i
$d^r_{0,i}$	Real distance between target node 0 and anchor node i
ϵ, $\overline{\epsilon}$	Localization errors for ASL and NSL models, respectively
δ_i	Measurement noise between target node 0 to anchor node i
e_i, f_i	Matrix decomposition terms $e_i = \sum_{j=1}^n x_{i,j}^2$, $f_i = d_{0,i}^2$

The localization process consists of three phases: anchor discovery, ranging, and 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 $(i = 0, 1, ..., m)$, denote its location by $\mathbf{x}_i = (x_{i,1}, x_{i,2}, ..., x_{i,n})$, where n is the dimensionality of the space, and $\mathbf{x}_0$ needs to be calculated.

2) Ranging: The distances between each anchor node and the target node are estimated. Two cases can be considered: anchor-ranging and 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 [17], [18], we only consider target-ranging-based localization in this work.

Ranging could be conducted through different ways, such as time-of-arrival (ToA)-based acoustic ranging [19] and radio frequency (RF) ranging [20]. Because we only focus on the privacy aspect of localization, improving the accuracy of ranging methods is not considered in this paper.

3) Location Computation: Based on the information of $(\mathbf{x}_i, d_{0,i})$, the multi-lateration method [21] 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.

B. Adjacent Subtraction based Localization

In order to securely calculate $\mathbf{x}_0$, the classical NSL model in [17], [18], [21] used a nonadjacent subtraction based approach to cancel the quadratic terms in the multi-lateration method. Specifically, each node i $(i = 1, ..., m)$ is supposed to satisfy a condition $\sqrt{\sum_{j=1}^n (x_{0,j} - x_{i,j})^2} = d_{0,i}$, where $x_{0,j}$ $(j = 1, ..., n)$ is the variable to be determined. To make the system easier for secure computation, the condition is rearranged as $\sum_{j=1}^n x_{0,j}^2 - 2\sum_{j=1}^n x_{0,j} x_{i,j} = d_{0,i}^2 - \sum_{j=1}^n x_{i,j}^2$. For the m such conditions, the quadratic term $\sum_{j=1}^n x_{0,j}^2$ is canceled by subtracting the mth equation by the ith one, getting a linear system $\mathbb{A}\mathbf{x}_0^T = \mathbb{B}$, where $\mathbb{A} = [A_1, ..., A_{m-1}]^T$, $\mathbb{B} = [b_1, ..., b_{m-1}]^T$,

$$A_i = 2[x_{m,1} - x_{i,1} \quad \dots \quad x_{m,n} - x_{i,n}], \tag{1}$$

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

The MMSE estimate for $\mathbb{A}\mathbf{x}_0^T = \mathbb{B}$ is given by $\widehat{\mathbf{x}}_0^T = (\mathbb{A}^T\mathbb{A})^{-1}\mathbb{A}^T\mathbb{B}$. Based on the NSL model, existing privacy-preserving localization algorithm in [17], [18] applied Pailliar homomorphic encryption to securely calculate $\mathbf{x}_0$.

In this work, an adjacent subtraction based approach is introduced to derive the ASL model which enables that the EPPL algorithm does not use homomorphic encryption technique. In particular, the quadratic term $\sum_{j=1}^n x_{0,j}^2$ is canceled by subtracting the ith equation by the adjacent $(i+1)$th one, obtaining a linear system $\mathbb{H}\mathbf{x}_0^T = \mathbb{Q}$, where $\mathbb{H} = [H_1, ..., H_{m-1}]^T$, $\mathbb{Q} = [q_1, ..., q_{m-1}]^T$,

$$H_i = 2[x_{i,1} - x_{i+1,1} \quad \ldots \quad x_{i,n} - x_{i+1,n}], \tag{3}$$

$$q_i = \sum_{j=1}^{n}(x_{i,j}^2 - x_{i+1,j}^2) - (d_{0,i}^2 - d_{0,i+1}^2). \tag{4}$$

The ASL model $\widehat{\mathbf{x}}_0^T = (\mathbb{H}^T\mathbb{H})^{-1}\mathbb{H}^T\mathbb{Q}$ is derived from the linear system $\mathbb{H}\mathbf{x}_0^T = \mathbb{Q}$ based on the MMSE estimation.

The ASL and NSL models are equivalent through showing the equivalence between $\mathbb{H}\mathbf{x}_0^T = \mathbb{Q}$ and $\mathbb{A}\mathbf{x}_0^T = \mathbb{B}$. In order to analyze the equivalence of the localization errors between the ASL and NSL models, let $\epsilon = (\epsilon_1, ..., \epsilon_n)$ and $\overline{\epsilon} = (\overline{\epsilon}_1, ..., \overline{\epsilon}_n)$ denote the localization errors, respectively. Let $\mathbf{x}_0^r = (x_{0,1}^r, ..., x_{0,n}^r)$ denote the real location of the target node. Using $\mathbf{x}_0^r + \epsilon$ to substitute $\mathbf{x}_0$ in $\mathbb{H}\mathbf{x}_0^T = \mathbb{Q}$, a linear system $\mathbb{H}\epsilon^T = \mathbb{C}$ can be derived, where ϵ is the unknown variable, $\mathbb{C} = [c_1, c_2, ..., c_{m-1}]^T$, and $c_i = \sum_{j=1}^n(x_{i,j}^2 - x_{i+1,j}^2) - (d_{0,i}^2 - d_{0,i+1}^2) - \sum_{j=1}^n(x_{i,j} - x_{i+1,j})x_{0,j}^r$. Similarly, a linear system $\mathbb{A}\overline{\epsilon}^T = \overline{\mathbb{C}}$ for the NSL model can be derived. Therefore, showing the equivalence between $\mathbb{H}\epsilon^T = \mathbb{C}$ and $\mathbb{A}\overline{\epsilon}^T = \overline{\mathbb{C}}$ is proving that the ASL model and the NSL model have equivalent localization error.

***Theorem** 1:* $\mathbb{H}\mathbf{x}_0^T = \mathbb{Q}$ and $\mathbb{A}\mathbf{x}_0^T = \mathbb{B}$ are equivalent and $\mathbb{H}\epsilon^T = \mathbb{C}$ and $\mathbb{A}\overline{\epsilon}^T = \overline{\mathbb{C}}$ are equivalent.

Proof: For the equivalence between $\mathbb{H}\mathbf{x}_0^T = \mathbb{Q}$ and $\mathbb{A}\mathbf{x}_0^T = \mathbb{B}$, the proof is to show that: i) an arbitrary solution of $\mathbb{H}\mathbf{x}_0^T = \mathbb{Q}$ is the solution of $\mathbb{A}\mathbf{x}_0^T = \mathbb{B}$; ii) an arbitrary solution of $\mathbb{A}\mathbf{x}_0^T = \mathbb{B}$ is the solution of $\mathbb{H}\mathbf{x}_0^T = \mathbb{Q}$.

Argument i) is proved in the following way. Let $\mathbf{x}_c = (x_{c,1}, ..., x_{c,n})$ denote an arbitrary solution of $\mathbb{H}\mathbf{x}_0^T = \mathbb{Q}$. So we can get $\mathbb{H}\mathbf{x}_c^T = \mathbb{Q}$. The ith equation is $2\sum_{j=1}^n(x_{i,j} - x_{i+1,j})x_{c,j} = \sum_{j=1}^n(x_{i,j}^2 - x_{i+1,j}^2) - (d_{0,i}^2 - d_{0,i+1}^2)$, which is changed as $2\sum_{j=1}^n x_{i,j}x_{c,j} - \sum_{j=1}^n x_{i,j}^2 + d_{0,i}^2 = 2\sum_{j=1}^n x_{i+1,j}x_{c,j} - \sum_{j=1}^n x_{i+1,j}^2 + d_{0,i+1}^2$. This means that all the left parts of the equation are equal, which means that $2\sum_{j=1}^n x_{i,j}x_{c,j} - \sum_{j=1}^n x_{i,j}^2 + d_{0,i}^2 = 2\sum_{j=1}^n x_{m,j}x_{c,j} - \sum_{j=1}^n x_{m,j}^2 + d_{0,m}^2$. Thus, the above equation can be changed as $2\sum_{j=1}^n(x_{m,j} - x_{i,j})x_{c,j} = \sum_{j=1}^n(x_{m,j}^2 - x_{i,j}^2) - (d_{0,m}^2 - d_{0,i}^2)$. The above equation is the linear system derived through subtracting the mth equation by the ith one, i.e., $\mathbb{A}\mathbf{x}_0^T = \mathbb{B}$, which means $\mathbf{x}_c$ is the solution of $\mathbb{A}\mathbf{x}_0^T = \mathbb{B}$.

Argument ii) can be proved in a similar way. Thus, the equivalence between $\mathbb{H}\mathbf{x}_0^T = \mathbb{Q}$ and $\mathbb{A}\mathbf{x}_0^T = \mathbb{B}$ is proved. For the equivalence between $\mathbb{H}\epsilon^T = \mathbb{C}$ and $\mathbb{A}\overline{\epsilon}^T = \overline{\mathbb{C}}$, the proof also has a similar way. Therefore, Theorem 1 is proved. ■

***Remark** 1:* Theorem 1 presents the equivalence between the ASL and NSL models by showing the equivalence between their corresponding linear systems. The difference between them is that they use adjacent and nonadjacent subtraction methods to cancel the quadratic terms $\sum_{j=1}^n x_{0,j}^2$, respectively. More importantly, based on the ASL model, one can develop an EPPL algorithm without using the homomorphic encryption. However, based on the NSL model, one cannot.

***Remark** 2:* Based on the equivalence analysis, the relationship between the localization error and the measurement noise is also derived. In particular, the MMSE estimate for $\mathbb{H}\epsilon^T = \mathbb{C}$ is given by $\epsilon^T = (\mathbb{H}^T\mathbb{H})^{-1}\mathbb{H}^T\mathbb{C}$. Let δ_i and $d_{0,i}^r$ denote the measurement noise and real distance between node 0 and node i $(i = 1, ..., m)$, respectively. Through substituting $d_{0,i}$ with $(d_{0,i}^r + \delta_i)$ in $\mathbb{C}$, the relationship model between the localization error ϵ and the measurement noise $\delta = (\delta_1, ..., \delta_m)$ is derived as $\epsilon^T = (\mathbb{H}^T\mathbb{H})^{-1}\mathbb{H}^T\mathbb{C}^{'}$, where $\mathbb{C}^{'} = [c_1^{'}, c_2^{'}, ..., c_{m-1}^{'}]^T$, and $c_i^{'} = \sum_{j=1}^n(x_{i,j}^2 - x_{i+1,j}^2) - [(d_{0,i}^r + \delta_i)^2 - (d_{0,i+1}^r + \delta_{i+1})^2] - \sum_{j=1}^n(x_{i,j} - x_{i+1,j})x_{0,j}^r$.

C. Problem Formulation

Existing work in [17], [18] utilized homomorphic encryption technique to propose a Privacy-Preserving Localization (PPL) algorithm based on the NSL model. However, under the ASL model, the EPPL algorithm needs to be designed without using the homomorphic encryption technique in this work. In particular, for target-ranging-based localization, node i has privacy concern on $\mathbf{x}_i$ and node 0 has privacy concern on $(\mathbf{x}_0, d_{0,i})$. The EPPL algorithm calculates $\mathbf{x}_0$ without allowing node $j \neq i$, where $j = 0, 1, ..., m$ and $i = 0, 1, ..., m$, to learn the private information $\mathbf{x}_i$. Meanwhile, $d_{0,k}$ is not allowed to learn by node k, where $k = 1, ..., m$. Moreover, the EPPL algorithm needs to be in a distributed way since mobile users are unwilling to reveal their location to a third party. Correctness, privacy, and efficiency of the EPPL algorithm also need to be analyzed.

III. Efficient Privacy-Preserving Localization

A. Basic Idea

A matrix decomposition method is used to securely calculate $\widehat{\mathbf{x}}_0^T = (\mathbb{H}^T\mathbb{H})^{-1}\mathbb{H}^T\mathbb{Q}$. In particular, H_i and q_i can be rewritten as $H_i = 2(\mathbf{x}_i - \mathbf{x}_{i+1})$, $q_i = (e_i - e_{i+1}) - (f_i - f_{i+1})$, 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 $(i = 1, ..., m)$. If we directly compute $\widehat{\mathbf{x}}_0$, it will leak privacy among nodes. Thus, to enable privacy-preserving localization, based on the rewritten forms, the intermediate terms $\mathbb{H}^T\mathbb{H}$ and $\mathbb{H}^T\mathbb{Q}$ can be decomposed as

$$\begin{aligned}\mathbb{H}^T\mathbb{H} = 4(&\sum_{i=1}^{m-1}\mathbf{x}_i^T\mathbf{x}_i + \sum_{i=1}^{m-1}\mathbf{x}_{i+1}^T\mathbf{x}_{i+1} \\ &- \sum_{i=1}^{m-1}\mathbf{x}_i^T\mathbf{x}_{i+1} - \sum_{i=1}^{m-1}\mathbf{x}_{i+1}^T\mathbf{x}_i),\end{aligned} \tag{5}$$

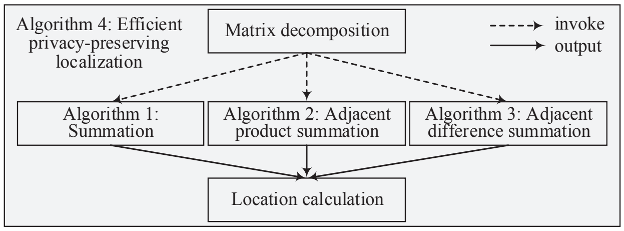
Algorithm 4: Efficient privacy-preserving localization
Matrix decomposition
invoke
output
Algorithm 1: Summation
Algorithm 2: Adjacent product summation
Algorithm 3: Adjacent difference summation
Location calculation

Figure 2. Algorithm relationship

$$\mathbb{H}^T\mathbb{Q} = 2[\sum_{i=1}^{m-1} e_i\mathbf{x}_i^T + \sum_{i=1}^{m-1} e_{i+1}\mathbf{x}_{i+1}^T - \sum_{i=1}^{m-1} e_i\mathbf{x}_{i+1}^T - \sum_{i=1}^{m-1} e_{i+1}\mathbf{x}_i^T - \sum_{i=1}^{m-1}(f_i - f_{i+1})(\mathbf{x}_i^T - \mathbf{x}_{i+1}^T)]. \tag{6}$$

An observation in above equations shows that $\sum_{i=1}^{m-1}\mathbf{x}_i^T\mathbf{x}_i$, $\sum_{i=1}^{m-1}\mathbf{x}_{i+1}^T\mathbf{x}_{i+1}$, $\sum_{i=1}^{m-1}e_i\mathbf{x}_i^T$, and $\sum_{i=1}^{m-1}e_{i+1}\mathbf{x}_{i+1}^T$ need to securely calculate the summation of $(m-1)$ nodes' private information. Moreover, $\sum_{i=1}^{m-1}\mathbf{x}_i^T\mathbf{x}_{i+1}$, $\sum_{i=1}^{m-1}\mathbf{x}_{i+1}^T\mathbf{x}_i$, $\sum_{i=1}^{m-1}e_i\mathbf{x}_{i+1}^T$, and $\sum_{i=1}^{m-1}e_{i+1}\mathbf{x}_i^T$ need to securely calculate the adjacent product summation of m nodes' private information. Note that $\sum_{i=1}^{m-1}\mathbf{x}_{i+1}^T\mathbf{x}_i$ is a matrix transposition of $\sum_{i=1}^{m-1}\mathbf{x}_i^T\mathbf{x}_{i+1}$ and we just calculate them once. Furthermore, $\sum_{i=1}^{m-1}(f_i - f_{i+1})(\mathbf{x}_i^T - \mathbf{x}_{i+1}^T)$ needs to securely calculate the adjacent difference summation of m nodes' private information. Note that this term should be calculated by node 0 since f_i is the private information of node 0. Therefore, in order to securely calculate $\widehat{\mathbf{x}}_0^T$, three privacy-preserving algorithms need to be designed for summation, adjacent product summation, and adjacent difference summation.

B. EPPL Algorithm

In this subsection, three sub-algorithms are first introduced. Then, the EPPL algorithm is described.

Figure 2 shows the relationship between the EPPL algorithm (Algorithm 4) and the three sub-algorithms (Algorithm 1, Algorithm 2, and Algorithm 3). In particular, the purposes of the three sub-algorithms are to securely calculate a summation, an adjacent product summation, and an adjacent difference summation, respectively. Based on the matrix decomposition, the EPPL algorithm invokes the three sub-algorithms to securely calculate the summation, the adjacent product summation, and the adjacent difference summation. With these outputs from the sub-algorithms, the EPPL algorithm securely calculates $\widehat{\mathbf{x}}_0^T$. The three sub-algorithms share the same idea to preserve privacy through random matrix/vector schemes. However these sub-algorithms are independent with each other since they have different purposes.

1) Privacy-Preserving Summation: Suppose node i has its private matrix M_i $(i = 1, ..., m)$ and node 0 wants to calculate the summation $\sum_{i=1}^{m} M_i$ without knowing any other nodes' private information. Algorithm 1 presents the privacy-preserving summation algorithm.

A random matrix scheme is used to secure the summation. In particular, each node i generates m random matrices such the summation of the matrices is zero. 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. Denote by P_i the constructed random matrix of node i. Note that $\sum_{i=1}^{m} P_i = 0$. Then, node i sends the mixed information of the random matrix and its private information to node 0. Without getting the real private information, node 0 computes the summation of all private information.

Algorithm 1 Summation

```
**Input**: node 0; $m$ nodes; node $i$'s private matrix $M_i$;
**Output**: $\alpha = \sum_{i=1}^{m} M_i$;
Each node $i$ $(i = 1, ..., m)$ generates $m$ random matrices $p_i^k$, where $k = 1, ..., m$, such that $\sum_{k=1}^{m} p_i^k = 0$ and $p_i^k$ has the same size as $M_i$. Node $i$ keeps one such matrix and sends the rest to the other $(m-1)$ nodes, respectively. Node $i$ constructs $P_i$ by adding up all $(m-1)$ matrices that it receives from other $(m-1)$ nodes with the one it keeps. Note that $P_i$ is a random matrix and $\sum_{i=1}^{m} P_i = 0$.
Node $i$ $(i = 1, ..., m)$ sends $\alpha_i = M_i + P_i$ to node 0.
Node 0 calculates $\alpha = \sum_{i=1}^{m} \alpha_i$ based on the received information.
```

2) Privacy-Preserving Adjacent Product Summation: Suppose node i has its private vector $V_i = [v_1^i, ..., v_n^i]$ $(i = 1, ..., m)$ and node 0 wants to calculate the adjacent production summation $\sum_{i=1}^{m}(V_i^T V_{i+1})$ without knowing any other nodes' 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

$$L_i = \begin{bmatrix} \log v_1^i & \cdots & \log v_n^i \\ \vdots & \ddots & \vdots \\ \log v_1^i & \cdots & \log v_n^i \end{bmatrix}. \tag{7}$$

Assume that all elements in V_i are positive (this assumption is correct since V_i denotes the coordinate of node i in this work). Algorithm 2 presents the privacy-preserving adjacent product summation algorithm. The random matrix scheme is used to secure the adjacent product summation in a similar way with that in Algorithm 1. Algorithm 2 always outputs the correct adjacent product summation without using homomorphic encryption. The logarithm function is used to transform the product into a summation so that it will cost less computation.

3) Privacy-Preserving Adjacent Difference Summation: Suppose node i has its private vector $V_i = [v_1^i, ..., v_n^i]$ $(i = 1, ..., m)$ and node 0 wants to calculate the adjacent difference summation $\sum_{i=1}^{m}(V_i^T - V_{i+1}^T)$ without knowing any other nodes' private information. Algorithm 3 presents the privacy-preserving adjacent difference summation algorithm. The random vector scheme is used in a similar way with that in Algorithm 1 to secure the adjacent difference summation without applying homomorphic encryption.

4) EPPL Algorithm: The EPPL algorithm is described in Algorithm 4 based on the three sub-algorithms. Suppose that node i has its privacy concern on $\mathbf{x}_i$ and e_i $(i = 1, ..., m)$.

Algorithm 2 Adjacent Product Summation

```
Input: node 0; $m$ nodes; node $i$'s private vector $V_i$;
Output: $P = \sum_{i=1}^{m} V_i^T V_{i+1}$;
For each node $i$ $(i = 1, ..., m-1)$, node $i$ and node $i+1$ do the following steps:
    i) Node $i$ and node $i+1$ construct the logarithm matrix $L_i$ and $L_{i+1}$ according to equation (7), respectively.
    ii) Node $i$ generates random matrices $w_i^i$ and $w_i^{i+1}$ with the same size as $L_i$ such that $w_i^i + w_i^{i+1} = 0$. Node $i$ keeps $w_i^i$ and sends $w_i^{i+1}$ to node $i+1$. Node $i$ constructs random matrix $\theta_i^{i+1} = w_i^i + u_{i+1}^i$ based on the kept matrix the received matrix. In a similar way, Node $i+1$ generates random matrices $u_{i+1}^{i+1}$ and $u_{i+1}^i$ and constructs random matrix $\theta_{i+1}^i = w_i^{i+1} + u_{i+1}^{i+1}$.
    iii) Node $i$ sends $\beta_i = L_i^T + \theta_i^{i+1}$ to node 0. Node $i+1$ sends $\beta_{i+1} = L_i + \theta_{i+1}^i$ to node 0.
Node 0 calculates $\beta_{i,i+1} = \beta_i + \beta_{i+1}$ based on the received information. Node 0 retrieves the product of elements from the matrix $\beta_{i,i+1}$ through an exponentiation operation and constructs the adjacent product $P_{i,i+1} = V_i^T V_{i+1}$. Then, node 0 computes $P = \sum_{i=1}^{m-1} P_{i,i+1}$.
```

Algorithm 3 Adjacent Difference Summation

```
Input: node 0; $m$ nodes; node $i$'s private vector $V_i$;
Output: $\gamma = \sum_{i=1}^{m} (V_i - V_{i+1})$;
For each node $i$ $(i = 1, ..., m-1)$, node $i$ and node $i+1$ do the following steps:
    i) Node $i$ generates random vectors $\varepsilon_i^i$ and $\varepsilon_i^{i+1}$ with the same size as $V_i$ such that $\varepsilon_i^i + \varepsilon_i^{i+1} = 0$. Node $i$ keeps $\varepsilon_i^i$ and sends $\varepsilon_i^{i+1}$ to node $i+1$. Node $i$ constructs random vector $\Delta_i^{i+1} = \varepsilon_i^i + \xi_{i+1}^i$ based on the kept vector and the received vector. In a similar way, Node $i+1$ generates random vectors $\xi_{i+1}^{i+1}$ and $\xi_{i+1}^i$ and constructs random vector $\Delta_{i+1}^i = \xi_{i+1}^{i+1} + \varepsilon_i^{i+1}$.
    ii) Node $i$ sends $\gamma_i = V_i + \Delta_i^{i+1}$ to node 0. Node $i+1$ sends $\gamma_{i+1} = -V_{i+1} + \Delta_{i+1}^i$ to node 0.
Node 0 calculates $\gamma_{i,i+1} = \gamma_i + \gamma_{i+1}$ based on the received information and calculates $\gamma = \sum_{i=1}^{m-1} \gamma_{i,i+1}$.
```

Node 0 has its privacy concern on $\widehat{\mathbf{x}}_0$ and f_i. Node 0 wants to securely calculate $\widehat{\mathbf{x}}_0$.

The input of Algorithm 4 is node i's location information $\mathbf{x}_i$, e_i which is based on its location information, and f_i which is based on 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.

Three nodes (e.g., the mth, $(m-1)$th, $(m-2)$th nodes) participate in the computation in Algorithm 4 in order that node 0 cannot calculate other nodes' location through solving an equation set. In particular, according to Algorithm 2, the three nodes separately need to calculate $\psi_1 = 4\sum_{i=1}^{m-1} \mathbf{x}_i^T \mathbf{x}_{i+1}$, $\psi_2 = 2\sum_{i=1}^{m-1} e_i \mathbf{x}_{i+1}^T$, and $\psi_3 = 2\sum_{i=1}^{m-1} e_{i+1} \mathbf{x}_i^T$, respectively. According to Algorithm 1, these nodes also separately calculate $\Omega_1 = 4\sum_{i=1}^{m-1} \mathbf{x}_i^T \mathbf{x}_i$, $\Omega_2 = 4\sum_{i=1}^{m-1} \mathbf{x}_{i+1}^T \mathbf{x}_{i+1}$, $\Omega_3 = 2\sum_{i=1}^{m-1} e_i \mathbf{x}_i^T$, and $\Omega_4 = 2\sum_{i=1}^{m-1} e_{i+1} \mathbf{x}_{i+1}^T$. Furthermore, the calculated results are sent to node 0. The minimum number of these participating nodes is studied in Theorem 4.

Node 0 first calculates $\phi = 2\sum_{i=1}^{m-1} (f_i - f_{i+1})(\mathbf{x}_i^T - \mathbf{x}_{i+1}^T)$ according to Algorithm 3. Then, it estimates its location through the matrix decomposition process.

Based on the matrix decomposition and the sub-algorithms, the EPPL algorithm always securely calculates the estimation of node 0's location without using the homomorphic encryption technique. Meanwhile, based on Theorem 1, the ASL model has equivalent localization error with the NSL model, which means that the proposed ASL based EPPL algorithm has equivalent accuracy with the NSL based privacy-preserving localization algorithm.

Algorithm 4 Efficient Privacy-Preserving Localization

```
Input: node $i$'s private information $\mathbf{x}_i$ and $e_i$; node 0's private information $f_i$; $i = 1, ..., m$;
Output: node 0's location estimation $\widehat{\mathbf{x}}_0$;
The $m$th node calculates $\Omega_1 = 4\sum_{i=1}^{m-1} \mathbf{x}_i^T \mathbf{x}_i$ and $\Omega_3 = 2\sum_{i=1}^{m-1} e_i \mathbf{x}_i^T$ according to Algorithm 1. The $m$th node calculates $\psi_1 = 4\sum_{i=1}^{m-1} \mathbf{x}_i^T \mathbf{x}_{i+1}$ according to Algorithm 2. Then, the $m$th node sends $\Omega_1$, $\Omega_3$, and $\psi_1$ to node 0.
The $(m-1)$th node calculates $\Omega_2 = 4\sum_{i=1}^{m-1} \mathbf{x}_{i+1}^T \mathbf{x}_{i+1}$ according to Algorithm 1. The $(m-1)$th node calculates $\psi_2 = 2\sum_{i=1}^{m-1} e_i \mathbf{x}_{i+1}^T$ according to Algorithm 2. Then, the $(m-1)$th node sends $\Omega_2$ and $\psi_2$ to node 0.
The $(m-2)$th node calculates $\Omega_4 = 2\sum_{i=1}^{m-1} e_{i+1} \mathbf{x}_{i+1}^T$ according to Algorithm 1. The $(m-2)$th node calculates $\psi_3 = 2\sum_{i=1}^{m-1} e_{i+1} \mathbf{x}_i^T$ according to Algorithm 2. Then, the $(m-2)$th node sends $\Omega_4$ and $\psi_3$ to node 0.
Node 0 calculates $\phi = 2\sum_{i=1}^{m-1} (f_i - f_{i+1})(\mathbf{x}_i^T - \mathbf{x}_{i+1}^T)$ according to Algorithm 3. Then, node 0 calculates $\widehat{\mathbf{x}}_0^T = (\Omega_1 + \Omega_2 - \psi_1 - \psi_1^T)^{-1}(\Omega_3 + \Omega_4 - \psi_2 - \psi_3 - \phi)$.
```

IV. Algorithm Analysis

In this section, the correctness, privacy and efficiency of the EPPL algorithm are analyzed.

A. Correctness Analysis

The correctness of the EPPL algorithm means that the EPPL algorithm is correct to calculate the target node's location. Specifically, $\widehat{\mathbf{x}}_0^T$ is correctly calculated by the EPPL algorithm based on the ASL model $\widehat{\mathbf{x}}_0^T = (\mathbb{H}^T\mathbb{H})^{-1}\mathbb{H}^T\mathbb{Q}$. Through the matrix decomposition, it is known that $\mathbb{H}^T\mathbb{H} = \Omega_1 + \Omega_2 - \psi_1 - \psi_1^T$ and $\mathbb{H}^T\mathbb{Q} = \Omega_3 + \Omega_4 - \psi_2 - \psi_3 - \phi$. Since the summations of the random matrices/vectors in the three sub-algorithms are zero when calculating Ω_1, Ω_2, Ω_3, Ω_4, ψ_1, ψ_2, ψ_3, ϕ, it is easy to obtain the following theorem.

***Theorem** 2:* Algorithm 4 correctly calculates the MMSE estimate $\widehat{\mathbf{x}}_0^T$ defined in $\mathbb{H}\mathbf{x}_0^T = \mathbb{Q}$.

B. Privacy Analysis

When the EPPL algorithm ends, it achieves privacy-preserving localization if the target node only knows its location and any anchor node does not learn other nodes' locations, including the target node's location.

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 computations 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: independent nodes and 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** 3:* For independent nodes, the EPPL algorithm achieves privacy-preserving localization when $m > 3$.

Proof: The proof is to show that: i) no anchor can learn the location of another anchor; ii) no anchor can learn the location of the target, not even obtain a rough estimation about the location of the target; and iii) the target cannot learn the location of any anchor.

Argument i) contains two cases: ordinary anchor nodes and the mth, $(m-1)$th, and $(m-2)$th anchor nodes that participate in the computation. The first case is obvious for anchor nodes $1, ..., m-3$ because the only information exchange between any two anchors i and j, where $1 \leq i, j \leq m$, is the random matrices/vectors which are not related to the location of any node. In particular, the random matrices/vectors in the sub-algorithms are used to generate P_i, $\{\theta_i^{i+1}, \theta_{i+1}^i\}$, and $\{\Delta_i^{i+1}, \Delta_{i+1}^i\}$, respectively.

For the second case, the only way that the mth anchor node can calculate other anchor nodes' location is to solve a linear equation set construct from the received information β_i, β_{i+1}, α_i^1, α_i^3, where $i = 1, ..., m-1$. By treating $\mathbf{x}_i$ $(i \neq m)$, θ_i, θ_{i+1}, P_i^1, P_i^3 as variables, the total number of the variables is $(m-1)n + 4(m-1)n^2$. The total number of independent linear equations the mth anchor node may obtain is at most $4(m-1)n^2 + (m-1) + 2$, which is obtained by knowing β_i, β_{i+1}, α_i^1, α_i^3, and the relationship $\theta_i + \theta_{i+1} = 0$, $\sum_{i=1}^{m-1} P_i^1 = 0$, $\sum_{i=1}^{m-1} P_i^3 = 0$. It can be seen that the number of variables is greater than the number of equations when $m > 3$. Therefore, the mth anchor node cannot calculate the location of any other anchor node when $m > 3$. In a similar way, it can be proved that the $(m-1)$th, and $(m-2)$th anchor nodes also cannot calculate the location of any other anchor node.

Argument ii) is true. From the matrix decomposition, it is clear that calculating the estimation $\widehat{\mathbf{x}}_0^T$, requires Ω_1, Ω_2, Ω_3, Ω_4, ψ_1, ψ_2, ψ_3, 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 the anchor node to guess the $\widehat{\mathbf{x}}_0^T$.

Argument iii) can be proved in the following way. In particular, the target can get more m equations based on the private ranging distances of the m anchors. Furthermore, there is an equation set constructed from knowing γ_i, γ_{i+1}, $\Delta_i + \Delta_{i+1} = 0$. However, there are variables from $\mathbf{x}_i$, Δ_i, Δ_{i+1}. Thus, the total number of variables is $mn + 2(m-1)n$. The total number of independent linear equations the target node may obtain is at most $m + 2(m-1)n + (m-1)$. It can be found that the number of the variables is greater than the number of the equations, and therefore the target node cannot calculate the location of any anchor node.

Combining the above arguments, Theorem 3 is proved. ■

The minimum number of the anchor nodes that participate in the computation (e.g., the mth, $(m-1)$th, and $(m-2)$th anchor nodes) is studied. These anchor nodes not only provide their locations, but also need to perform some computation. If there do not exist these anchor nodes and just let the target node perform all the computation, it is possible that the target node learns all anchor nodes' location through solving an equation set based on the received information.

***Theorem** 4:* For independent nodes, the EPPL algorithm achieves privacy-preserving localization if and only if at least three anchor nodes separately calculate ψ_1, ψ_2, ψ_3, and they also separately calculate $\Omega_1, \Omega_2, \Omega_3, \Omega_4$, when $m > 3$.

Proof: The proof is to show that: i) at least three anchor nodes ⇒ privacy-preserving localization; ii) privacy-preserving localization ⇒ at least three anchor nodes.

Argument i) is proved as follows. Theorem 3 has proved that three anchor nodes participate in the computation to ensure privacy-preserving localization. If the number of the participating anchor nodes is more than three, the additional anchor nodes can be used to perform calculations in order to decrease the computation cost of the mth, $(m-1)$th, and $(m-2)$th anchor nodes in Algorithm 4.

Argument ii) is proved as follows. Privacy-preserving localization needs to securely calculate Ω_1, Ω_2, Ω_3, Ω_4, ψ_1, ψ_2, ψ_3, ϕ. Since f_i belongs to the target, ϕ must be calculated by the target. Three cases need to be considered: the target cannot additionally calculate other term; private information will leak if only two anchors participate in the calculation; three anchors must separately calculate ψ_1, ψ_2, ψ_3, and separately calculate $\Omega_1, \Omega_2, \Omega_3, \Omega_4$. First, as discussed in Argument iii) in the proof of Theorem 3, if the target calculates an additional term, it will have a chance to calculate the location of some anchors. Second, suppose that there are only two anchors that participate in the calculation. In order to calculate ψ_1, ψ_2, ψ_3, one anchor at least needs to perform calculation for two terms (let's say ψ_1, ψ_2). This node has a chance to formulate an equation set and learns other anchors' location. Third, suppose that there are three anchors that participate in the calculation. As discussed in the second case, three anchors must separately calculate ψ_1, ψ_2, ψ_3. If one anchor calculates both one adjacent product summation (let's say ψ_1) and at most two summations (let's say Ω_1, Ω_3), it is easy to prove that the anchor cannot learn other nodes' location. Otherwise, private information leaks. Thus, Argument ii) is correct.

Combining the above arguments, Theorem 4 is proved. ■

***Theorem** 5:* For colluding nodes, the EPPL algorithm achieves privacy-preserving localization if node 0 and the mth, $(m-1)$th, $(m-2)$th nodes do not involve in the collusion.

Proof: Since both the target node and the three participating anchor nodes do not involve in the collusion, the proof is to show that i) the colluding anchor nodes cannot obtain a rough estimation about $\widehat{\mathbf{x}}_0^T$ and ii) the colluding anchor nodes cannot calculate any other anchor nodes' location.

Argument i) is correct because the EPPL algorithm requires the target node to perform the ranging process. The target node takes ranging results as its private information and each anchor node only has its own location information. The colluding anchor nodes cannot construct a small-scale multi-lateration linear system to estimate the target node's location.

Argument ii) can be proved by noting that the only information exchange between any two anchors i and j, where $1 \leq i, j \leq m$, is the random matrices/vectors which are not related to location information. Thus, the colluding anchor nodes cannot calculate any other anchor nodes' location.

Combining the above arguments, Theorem 5 is proved. ■

***Remark** 3:* The above theorems are analyzed as follows. Theorem 2 guarantees that the EPPL algorithm correctly calculates the target node's location. Theorem 3 and Theorem 5 prove that the EPPL algorithm achieves privacy-preserving localization for both independent and colluding scenarios, respectively, which means the privacy can be preserved when performing the EPPL algorithm. Theorem 4 studies the minimum number of the participating anchors in independent scenario and proves that the condition of three participating anchor nodes is the sufficient and necessary condition for the EPPL algorithm to perform privacy-preserving localization.

***Remark** 4:* In terms of privacy/performance tradeoff analysis, the EPPL algorithm is compared with the PPL algorithm which is based on homomorphic encryption. For independent nodes, both the EPPL algorithm and the PPL algorithm achieve privacy-preserving localization. The condition for the EPPL algorithm is that it requires $m > 3$, while the condition for the PPL algorithm is that it requires $m > n$. For colluding nodes, both algorithms achieve privacy-preserving localization under different conditions. The EEPL algorithm requires the target node and three computation-participating anchor nodes do not involve in the colluding, while the PPL algorithm requires that the number of colluding anchor nodes is less than half of $(m-1)$ and the number of of noncolluding anchor nodes is greater than $(n+1)$. In conclusion, the EPPL algorithm achieves privacy-preserving localization without using the homomorphic encryption technique.

C. Efficiency Analysis

The computation and communication overheads of the EPPL algorithm are analyzed. The efficiency analysis follows the methodology in [17], [18], where the computation overhead is dominated by the vector operations and the communication overhead is dominated by the transmission of elements.

1) Computation Overhead: The computation overhead is analyzed on both the anchor node and the target node sides.

The computation overhead on each anchor node is examined as follows. First, for the anchor nodes that do not participate in the computation (let's say ordinary anchor nodes), their computation overhead is dominated by vector multiplication and logarithm operations when performing the sub-algorithms. The total computation overhead on an ordinary anchor node is (n^2+2n) logarithm operations and (n^2+n) multiplication operations. Then, for the three anchor nodes that participate in the computation (let's say special anchor nodes), their computation overhead is dominated by not only the vector multiplication and logarithm operations in the sub-algorithms, but also the exponentiation operations in Algorithm 2. Thus, the total computation overhead on a special anchor node is (n^2+2n) logarithm operations, (n^2+n) multiplication operations, and n^2 exponentiation operations.

The computation overhead on the target node is examined through analyzing vector multiplication operations and matrix inversion operations. In particular, the target node needs to perform $(m-1)n$ multiplications for calculating ϕ, one matrix inversion operation (n^4 multiplication), and n^3 multiplications in Step 4 of Algorithm 4. Therefore, the total computation overhead is $(m-1)n+n^3+n^4$ multiplications.

2) Communication Overhead: The communication overhead is analyzed for the EPPL algorithm. In particular, the communication overhead is dominated by the element transmission in Algorithm 4. An anchor node i $(i = 1, 2, ..., m-1)$ needs to transmit $n+n^2+n^2$ elements to the target node and an anchor node i $(i = 1, 2, ..., m-3)$ transmits $n+n^2+n^2$ elements to the mth, $(m-1)$th, $(m-2)$th anchor nodes in the sub-algorithms. In Step 1 of Algorithm 4, the mth anchor node needs to transmit n^2+n+n^2 elements to the target node. In Step 2 of Algorithm 4, he $(m-1)$th anchor node needs to transmit $n+n^2$ elements to the target node. In Step 3 of Algorithm 4, the $(m-2)$th anchor node needs to transmit $n+n$ elements to the target node. Thus, the total number of elements is $4mn^2+2mn-3n^2+2n$.

V. Performance Evaluation

In this section, we evaluate the performance of the ASL model and the EPPL algorithm with simulations and numerical results.

A. Simulations

In order to compare the performance between the ASL model and the NSL model, some simulations are conducted using Matlab software which runs on a DELL desktop PC with Intel(R) core (TM) 2 Quad CPU Q9450 @ 2.66Hz processor and 4.00GB (3.24 GB available) RAM. We consider a 500m×500m square area. The target node is located at the center of the square and the anchor nodes are uniformly randomly distributed in the area. The ranging distance is simulated as the real distance ($d_{0,i}^r$) plus a ranging error $d_{0,i} = d_{0,i}^r + \delta_i$. The distribution for the ranging error δ_i is a Gaussian distribution with mean $\mu = 0$ and standard deviation $\sigma = 5$. For each simulation, we performed 1000 independent runs and reported the average of the results in order to reduce

Table II
COMPUTATION AND COMMUNICATION OVERHEADS COMPARISONS

	EPPL	PPL
Computation on each ordinary anchor node	$(n^2+2n)\varepsilon_1+(n^2+n)\chi_1$	$2\varepsilon_2+2n\varepsilon_3+n\chi_2+(3n^2+4n)\chi_1$
Computation on each special anchor node	$(2n^2+2n)\varepsilon_1+(n^2+n)\chi_1$	
Computation on the target node	$[n^4+n^3+(m-1)n]\chi_1$	$m\chi_2+m\varepsilon_3+(n^4+n^3)\chi_1$
Communication	$[4mn^2+2mn-3n^2+2n]\times 24$	$2048mn+[m^2n^2+(3n+1)m^2]\times 24$

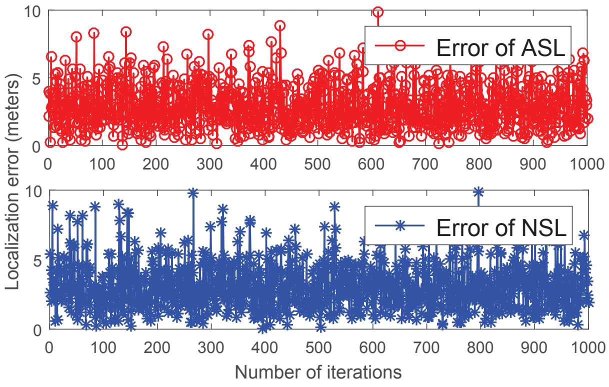

Figure 3. Localization errors with the number of iterations

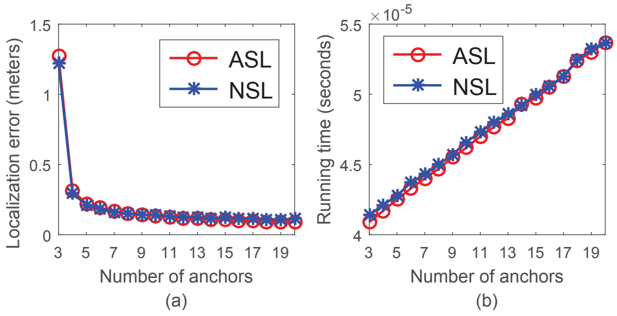

Figure 4. Localization error and running time

the randomness of the random variable. When computing the localization error, we use $LocError = \|\bar{\mathbf{x}}_0 - \mathbf{x}_0^r\|$, where $\bar{\mathbf{x}}_0 = (\sum_{k=1}^N \widehat{\mathbf{x}}_0^k)/N$, N denotes the number of independent runs, $\widehat{\mathbf{x}}_0^k$ is the estimated target location of the kth run, and $\mathbf{x}_0^r$ is the target node's real location.

Figure 3 presents that the localization error with the increasing number of iterations. The circle nodes and the star nodes denote the localization errors of the ASL model and the NSL model, respectively. Both the localization errors of the models are similar since the two models are equivalent. In particular, most of the localization errors are from 0 to 5 meters.

Figure 4 illustrates the localization error and running time with the increasing number of anchors. The running time is an average of the time over 1000 independent runs in order to show the running time of a single simulation. It can be found that, in Figure 4(a), as the given area is 500m×500m square area, most of the localization errors (less than 0.2 meters) of both localization models are close enough because of the equivalence between them. Furthermore, in Figure 4(b), it can be seen that both localization models have a similar running time also because of their equivalence.

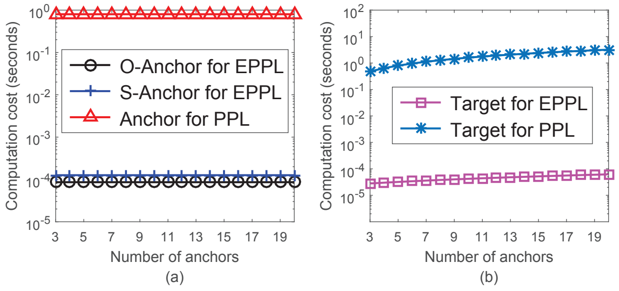

Figure 5. Computation cost

From the above simulations, one sees that the ASL and NSL models have similar performance even though they are based on different subtraction methods to cancel the quadratic terms. At the same time, the ASL model can be used to design the EPPL algorithm without applying the homomorphic encryption technique while the NSL model cannot.

B. Numerical Results

Table II presents the comparisons of the computation and communication overheads between the EPPL algorithm and the PPL algorithm. Following the setting in [17], [18], we assume that a real number is represented by 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: $\chi_1 = 1\mu$s, $\chi_2 = 0.88$ms, $\varepsilon_1 = 10\mu$s, $\varepsilon_2 = 81.08$ms, and $\varepsilon_3 = 159.06$ms. The setting of these parameters also follows [17], [18], which are obtained from real experiments in [22]. We also assume the communication between nodes has a bandwidth of 2 Mb/s. Performance metrics include the total computation time, the total number of transmitted bits, and the algorithm execution time.

We plot the anchor node's computation cost as a function of the number of anchor nodes in Figure 5(a). In particular, the computation cost on each anchor of both the EPPL and PPL algorithms changes little with the number of anchors because the computation cost on the anchor side in both algorithms is just based on the algorithms and independent of the number

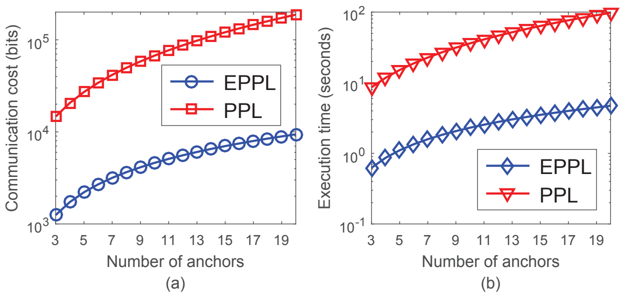

Figure 6. Communication cost and execution time

of anchor nodes. The computation cost of the special anchor (S-Anchor) node is a bit higher than that of the ordinary anchor (O-Anchor) node because the special anchor nodes (the mth, $(m-1)$th, $(m-2)$th nodes) participate in the computation. However, the computation cost of each anchor node in the homomorphic encryption based PPL algorithm is much higher than that in the EPPL algorithm because of the long-bit multiplications and exponentiations required by the encryption/decryption. Hence, each anchor node in EPPL algorithms has small computation cost, which indicates that it can attract more users to participate in the localization process in order to improve the accuracy of the localization result.

The target node's computation cost is plotted in Figure 5(b). The target node in the EPPL algorithm are much more computationally efficient than that in the PPL algorithm. Specifically, the EPPL algorithm reduces the total CPU time by at least 4 orders of magnitude when compared to the PPL algorithm. This is because that the PPL algorithm is based on the homomorphic encryption process.

The communication cost and the execution time of the EPPL and PPL algorithms are compared in Figure 6. In particular, it can be seen that in Figure 6(a) the communication cost of the EPPL algorithm is much smaller than that of the PPL algorithm because its calculation does not need the homomorphic encryption process. In Figure 6(b), the execution time of the EPPL algorithm is smaller than that of the PPL algorithm because of no homomorphic encryption process. Specifically, the execution time of the EPPL algorithm ranges from hundreds of milliseconds to five seconds.

VI. Conclusion

In this paper, to enable efficient privacy-preserving, an adjacent subtraction based localization model ASL is used to replace the nonadjacent subtraction based localization model NSL with acceptable localization error and running time. An efficient privacy-preserving localization algorithm EPPL is proposed without using the time-consuming homomorphic encryption. Correctness analysis, privacy analysis, efficiency analysis, numerical results, and some simulations are presented to evaluate the ASL model and the EPPL algorithm.

Acknowledgments

This research is supported by China Scholarship Council; Future Network Foresight Program of Jiangsu Province (No. BY2013095-1-08); the College Graduate Innovation Projects of Jiangsu Province (No. CXLX12_0487); NUPTSF (No. NY215019); National Natural Science Foundation of China (No. 61502246); in part by NSERC, CFI and BCKDF.

References

[1] Foursquare, https://foursquare.com/

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

[3] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: indoor location sensing using active RFID," *Wireless networks*, vol. 10, pp. 701-710, 2004.

[4] Z. Yang and Y. Liu, "Quality of Trilateration: Confidence-based Iterative Localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, pp. 631-640, 2010.

[5] I. Guvenc, C. C. Chong, "A Survey on TOA Based Wireless Localization and NLOS Mitigation Techniques," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp.107-124, 2009.

[6] K. G. Shin, J. Xiaoen, C. Zhigang, and H. Xin, "Privacy protection for users of location-based services," *IEEE Wireless Communications*, vol. 19, no. 1, pp. 30-39, 2012.

[7] 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.

[8] 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*, 2015, pp. 288-297.

[9] H. Li, Y. He, X. Cheng, H. Zhu, and L. Sun, "Security and privacy in localization for underwater sensor networks," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 56-62, 2015.

[10] F. Buccafurri, G. Lax, S. Nicolazzo, and A. Nocera, "A Privacy-Preserving Localization Service for Assisted Living Facilities," *IEEE Transactions on Services Computing*, vol. PP, no. 99, pp. 1-1, 2016.

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

[12] X. Wang, Y. Liu, Z. Shi, X. Lu, and L. Sun, "A Privacy-Preserving Fuzzy Localization Scheme with CSI Fingerprint," in *Proc. IEEE GLOBECOM*, 2015, pp. 1-6.

[13] P. Armengol, R. Tobkes, K. Akkaya, S. B, iftler, G. I, "Efficient Privacy-Preserving Fingerprint-Based Indoor Localization Using Crowdsourcing," in *Proc. IEEE MASS*, 2015, pp. 549-554.

[14] A. Konstantinidis, G. Chatzimilioudis, D. Zeinalipour-Yazti, P. Mpeis, N. Pelekis, and Y. Theodoridis, "Privacy-preserving indoor localization on smartphones," in *Proc. IEEE ICDE*, 2016, pp. 1470-1471.

[15] A. Konstantinidis, G. Chatzimilioudis, D. Zeinalipour-Yazti, P. Mpeis, N. Pelekis, and Y. Theodoridis, "Privacy-Preserving Indoor Localization on Smartphones," *IEEE Transactions on Knowledge and Data Engineering*, vol.27, no. 11, pp. 3042-3055, 2015.

[16] S. U. Hussain, F. Koushanfar, "Privacy preserving localization for smart automotive systems," in *Proc. ACM/EDAC/IEEE DAC*, 2016, pp. 1-6.

[17] S. Tao, C. Yingying, Y. Jie, and A. Williams, "Multi-lateral privacy-preserving localization in pervasive environments," in *Proc. IEEE INFOCOM*, 2014, pp. 2319-2327.

[18] S. Tao, C. Yingying, and Y. Jie, "Protecting Multi-Lateral Localization Privacy in Pervasive Environments," *IEEE/ACM Transactions on Networking*, vol. 23, no. 5, pp. 1688-1701, 2015.

[19] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, "Push the limit of WiFi based localization for smartphones," in *Proc. ACM Mobicom*, 2012, pp. 305-316.

[20] 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.

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

[22] R. Zhang, J. Zhang, Y. Zhang, J. Sun, and G. Yan, "Privacy-Preserving Profile Matching for Proximity-Based Mobile Social Networking," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 656-668, 2013.