Multi-lateral Privacy-Preserving Localization in Pervasive Environments

Tao Shu*, Yingying Chen†, Jie Yang*, and Albert Williams‡
* Department of CSE, Oakland University, {shu, yang}@oakland.edu
† Department of ECE, Stevens Institute of Technology, yingying.chen@stevens.edu
‡ Department of CS, University of Massachusetts Amherst, albert.b.p.williams@gmail.com

***Abstract*—Location based services (LBSs) have raised serious privacy concerns in the society, due to the possibility of leaking a mobile user's location information in enabling location-dependent services. While existing location-privacy studies are mainly focused on preventing the leakage of user's location in accessing the LBS server, the possible privacy leakage during the localization process has been largely ignored. Such a privacy leakage stems from the fact that a localization algorithm typically takes the location of anchors (i.e., reference points for localization) as input, and generates the target's location as output. As such, the location of anchors, and consequently the target's location, could be leaked to others. An adversary could further utilize the leakage of anchor's locations to attack the localization infrastructure and undermine the accurate estimation of the target's location. To address this issue, in this paper, we study the *multi-lateral* privacy preserving localization problem, whereby the location of a target is calculated without the need of revealing anchors' location, and the knowledge of the localization outcome is strictly limited to the target itself. To fully protect user's privacy, our study protects not only the user's exact location information (the geo-coordinates), but also any side information that may lead to a coarse estimate of the location. Three privacy-preserving localization solutions are developed by leveraging combinations of information hiding and homomorphic encryption. These solutions provide different levels of protection for location side information and resilience to node collusion, and have the advantage of being able to trade user's privacy requirements for better computation/communication efficiency.**

I. Introduction

With the proliferation of location based services (LBSs), the issue of location privacy has raised serious concerns in the society. In LBS, a mobile user first obtains its location information from a localization infrastructure, and then uses this information to obtain location-dependent services from a LBS server. While the mobile user can enjoy the convenience brought by LBS, it is enticed to reveal its location to enable and receive the service, leading to potential leakage of the user's privacy. There have been extensive location-privacy studies focused on preventing a LBS server from learning a user's location when the user accesses the server with his location information, e.g., the k-anonymity [11], the mix zones [1], and the m-unobservability [5]. While these measures prevent location leakage in accessing the LBS server, they are carried out after the location has been calculated and obtained by the user, and thus have largely overlooked possible location leakage originated from the calculation of the location, i.e., *the localization process*.

In particular, privacy leakage in the localization process stems from the fact that a localization algorithm typically calculates a target's location based on the known location of several reference points (a.k.a. anchors) and the ranging information between the anchors and the target. Because the algorithm takes anchors's locations as input, and generates the target's location as output, multi-sided privacy leakage can happen. On one side, anchors have to reveal their location information, rendering such information potentially learnable by other nodes. This could lead to severe security issues. For instance, in WiFi localization an adversary can attenuate the signals from the Access Points (APs) by making use of the leaked AP's location information and attack the localization infrastructure (e.g., location spoofing attack) [25], [27], [16]. On the other side, as the outcome of the algorithm, the knowledge of the target's location may not be limited to the target itself. For example, the assisted-GPS (AGPS) system widely employed in today's smartphones relies on networked servers to calculate the location. As a result, the location of the user is also known by these servers.

978-1-4799-3360-0/14/$31.00 © 2014 IEEE

While existing research on the localization process are mainly focused on the algorithm's accuracy and energy efficiency, the privacy aspect during the localization process has been largely ignored. There are only few studies [24], [18], [30], [2] relying on special hardware such as antenna arrays to preserve the unilateral privacy aspect in the localization process, i.e, an anchor cannot learn the target's location, whereas the target can still obtain the anchors' location information. However, none of the existing studies have investigated the privacy leakage issue from the aspect of the anchors' location, which becomes more severe in the increasingly pervasive wireless environments. For instance, during the crowdsourcing-based localization [20], [23], GPS-enabled smartphones serve as ad hoc mobile anchors (a.k.a. *helpers*) to locate wireless devices (e.g., sensors or tablets) that do not own a traditional localization capability (GPS or cellular). However, these helpers' user-sensitive location information have been disclosed to the target object. Meanwhile, the target also considers the helpers as untrusted, and definitely does not want them to know the localization outcome, even though it needs them to participate during the localization process. Therefore, there is an urgent need to seek localization solutions that can address the privacy issues during the localization process itself by considering both the target object and the anchor points simultaneously.

Toward this end, in this paper we develop privacy-preserving localization algorithms by considering the privacy issues during the localization process. In particular, we study the more general *multi-lateral* privacy preservation problem, whereby the location of a target is calculated without the need of revealing anchors' location, and the knowledge of the localization outcome is strictly limited to the target itself. In other words, the location information of every node, including not only the target but also the anchors, is considered as private information of that node and is protected against every other node.

Our approach does not rely on specialized hardware. We study the privacy-preserving localization problem under a distributed setup, i.e., participants of localization are restricted to anchor points (including both public anchors and ad hoc anchor helpers) and the target. And the multi-lateral privacy preservation solution is more critical for scenarios using ad hoc anchor helpers (e.g., smartphones). The problem is trivial under a centralized setup, if there exists a third party trusted by all anchors and target. However, similar to the privacy argument frequently raised for LBS, we believe that mobile users who are concerned about revealing their location to LBS servers will likely be hesitant to entrust their location data to a third-party server. This further motivates us to seek a distributed solution to the problem.

One important feature of our privacy-preserving localization solution is that it develops unique three-level privacy protection and thus has the capability to protect any side information that may lead to a coarse estimate of the location in addition to the protection of the exaction location of the target. The side information during the localization process could include not only the anchor points' location information but also any intermediate result, which is a function of the locations, e.g., the relative ranging result between the target and the anchor. Such side information can usually lead to a coarse estimate of the target, which may be sufficient to reveal a large amount of privacy about the user. For instance, in a hospital, with a location resolution of a few ten meters, the adversary will be able to identify the department a mobile user is visiting, so it can conjecture the specific health problem the mobile user is having. Note that the requirement of protecting location side information is much stronger than a regular data privacy-preservation problem, e.g., those modeled by a classical multi-party secure computation problem [28], [10], [8], whose main goal is just to hide the value of the data.

To the best of our knowledge, our work is the first to provide a full range of privacy-overhead-balanced constructions to address the privacy issues during the localization process. Our contribution in this paper is three-fold.

- We propose and formulate the multi-lateral privacy preserving localization problem as a secure least-squared-error (LSE) estimation for an over-determined linear system. Different from other applied secure computation work that mainly deals with computation between two parties, e.g., inner product between two private vectors [7], [29], our problem concerns private parameters owned by multiple parties (corresponding to anchors and the target). Existing solutions to general secure LSE problems are based on oblivious transfer or homomorphic encryption, which typically have high computation complexity, and is originally designed for two parties only. A straightforward extension to multi-party computation will lead to overwhelming computation and communication overhead.
- Rather than relying on straightforward extension of existing solutions, we exploit the special structure of our problem to develop low-cost solutions. In particular, we define three levels of privacy, and develop efficient solutions for each of them using combinations of information hiding and homomorphic encryption techniques. These solutions have the benefit of being able to trade user's privacy requirements for better computation and communication efficiency, which is especially important in a resource-constrained mobile computing environment.
- We prove the privacy property for the proposed constructions and evaluate their computation/communication overhead using analysis and numerical methods. By comparing with existing LSE solutions, we verify the significant efficiency improvement of the proposed solutions.

The remainder of the paper is organized as follows. We define the system model and formulate the problem in Section II. The proposed privacy-preserving localization protocols are presented in Section III. Section IV evaluates the performance of the proposed mechanisms. Related work is reviewed in Section V and we conclude our work in Section VI.

II. System Model and Problem Formulation

A. System Model

We consider a general localization scenario where both the anchors and the target could be either static or mobile. Without loss of generality, we use the crowdsourcing-based localization as an example where both the anchors and the target are mobile. The localization session involves with multiple anchor helpers (e.g., smartphones) and one target mobile device (e.g., laptop, sensor, or tablet), denoted as node 0, and consists of three phases: anchor discovery, ranging, and location computation. In the first phase, node 0 recruits mobile anchors by broadcasting hello messages on all its communication interfaces. A smartphone receiving the hello message replies to node 0 to become an anchor. An anchor needs to satisfy the following two conditions: (1) It needs to be within one-hop communication distance from the target, so that some type of ranging can be performed. This means that the anchor is in the same cell as node 0 if a cellular interface is used, or in the same basic service set (BSS) if a WiFi interface is used. This condition is usually satisfied, because the anchor can receive the hello message in the first place. (2) The anchor must have the knowledge of its location. Node 0 may optionally indicate in the hello message a desired level of accuracy for anchor's location information (e.g., GPS-enabled). Only those anchors that satisfy this condition will reply. Let the number of anchors collected by node 0 be m, and denote them as nodes 1 to m, respectively. For node i, $i = 0, \ldots, m$, denote its location by $\mathbf{x}_i \overset{\text{def}}{=} (x_{i1}, \ldots, x_{in})$, where n is the dimensionality of the space ($n = 2$ for 2-D localization), and $\mathbf{x}_0$ is to be computed.

In the ranging phase, each anchor estimates its distance to the target. Let this distance estimate be d_{0i} for node $i = 1, \ldots, m$. Ranging could be based on various methods. For example, if an anchor and the target are in the same BSS, time-of-arrival (ToA) based acoustic ranging is possible, which allow the anchor to accurately measure d_{0i}, as experimented in [17]. On the other hand, if the separation between the anchor and the target is large, RF ranging will be used. The problem of improving the accuracy of various ranging methods is out of the scope of this work, as we are mainly focused on the privacy aspect of the localization process. Our constructions do not depend on the selection of ranging methods.

We use the method of multi-lateration for location calculation [21], due to its simplicity and popularity. In particular, based on $(\mathbf{x}_i, d_{0i})$'s, $i = 1, \ldots, m$, the multi-lateration method calculates the target location by minimizing the mean squared error (MMSE) between the measured distances (obtained in the ranging phase) and the calculated distances (based on location estimates). More specifically, every node $i = 1, \ldots, m$ is supposed to satisfy the following condition, respectively:

$$\sqrt{\sum_{j=1}^{n} (x_{0j} - x_{ij})^2} = d_{0i}, \quad i = 1, \ldots, m \tag{1}$$

where x_{0j}'s are variables to be resolved for the target location estimation. Because this is an over-decided system ($m > n$) and there are errors in the measurement of d_{0i}'s, it is unlikely that all above equations can be satisfied. So multi-lateration method estimates the target location $(\hat{x}_{01}, \ldots, \hat{x}_{0n})$ by minimizing the following mean square error

$$(\hat{x}_{01}, \ldots \hat{x}_{0n}) = \text{argmin}_{\mathbf{x}_0} \sum_{i=1}^{m} \left[\sqrt{\sum_{j=1}^{n} (x_{0j} - x_{ij})^2} - d_{0i} \right]^2 \tag{2}$$

B. Problem Statement: Privacy-Preserving Location Calculation

The system defined by condition (1) is quadratic. Little is known regarding the secure computation of its MMSE estimation defined in (2). To make the system more amenable to secure computation, we linearize it using the method described in [21]. In particular, (1) can be rewritten as

$$\sum_{j=1}^{n} x_{0j}^2 - 2 \sum_{j=1}^{n} x_{0j} x_{ij} = d_{0i}^2 - \sum_{j=1}^{n} x_{ij}^2 \quad i = 1, \ldots, m \tag{3}$$

For m such equations, the quadratic term $\sum_{j=1}^{n} x_{0j}^2$ can be canceled by subtracting the mth equation by the ith one $(i = 1, \ldots, m-1)$, getting the following derived linear system $\mathbf{A}\mathbf{x}_0^T = \mathbf{b}$, where

$$\mathbf{A} \overset{\text{def}}{=} 2 \begin{bmatrix} x_{m1} - x_{11} & \cdots & x_{mn} - x_{1n} \\ x_{m1} - x_{21} & \cdots & x_{mn} - x_{2n} \\ \vdots & \ddots & \vdots \\ x_{m1} - x_{m-11} & \cdots & x_{mn} - x_{m-1n} \end{bmatrix} \tag{4}$$

$$\mathbf{b} \overset{\text{def}}{=} \begin{bmatrix} \sum_{j=1}^{n} (x_{mj}^2 - x_{1j}^2) - (d_{0m}^2 - d_{01}^2) \\ \sum_{j=1}^{n} (x_{mj}^2 - x_{2j}^2) - (d_{0m}^2 - d_{02}^2) \\ \vdots \\ \sum_{j=1}^{n} (x_{mj}^2 - x_{m-1j}^2) - (d_{0m}^2 - d_{0m-1}^2) \end{bmatrix}. \tag{5}$$

Instead of solving (2), we focus on the derived linear system, because its linear nature is more amenable to secure computation. The MMSE estimate for this system is given by

$$\hat{\mathbf{x}}_0^T = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}. \tag{6}$$

An observation of the definition of $\mathbf{A}$ and $\mathbf{b}$ in (4) and (5) reveals that normally calculating $\hat{\mathbf{x}}_0^T$ requires nodes $i = 1, \ldots, m$ to disclose their $(\mathbf{x}_i, d_{0i})$'s to the algorithm.

Now suppose nodes $i = 0, 1, \ldots, m$ have privacy concern on $(\mathbf{x}_i, d_{0i})$'s and consider it as their private information. The problem of privacy-preserving location calculation is to design protocols to calculate (6) in such a way that the calculation does not allow any node $j \neq i$, where $j = 0, \ldots, m$ and $i = 0, \ldots, m$, to learn information on $(\mathbf{x}_i, d_{0i})$. Due to the reason highlighted in Section I, we are interested in distributed protocols whose calculation only involves nodes $i = 0, \ldots, m$. Note that protecting node i's location privacy means more than just hiding $\mathbf{x}_i$ from other nodes, as a node j may be able to compute an estimate of $\mathbf{x}_i$ based on some intermediate results of the calculation, if the protocol is not properly designed. In particular, depending on the amount of information leakage that can be tolerated, we define the following three levels of privacy (For ease of notation, but without leading to ambiguity, hereafter $\mathbf{x}_0$ and its MMSE estimation as defined in (6) are used interchangeably).

Definition 1. Level-I Privacy: When the protocol ends, node 0 knows $\mathbf{x}_0$. A node i, where $i = 0, \ldots, m$, will not know $\mathbf{x}_j$ for $\forall j = 0, \ldots, m, j \neq i$. However, a node $i \neq 0$ can compute by itself a coarse estimation of $\mathbf{x}_0$.

Definition 2. Level-II Privacy: When the protocol ends, node 0 knows $\mathbf{x}_0$. A node i, where $i = 0, \ldots, m$, will not know $\mathbf{x}_j$ for $\forall j = 0, \ldots, m, j \neq i$. However, a node $i \neq 0$ can compute a coarse estimate of $\mathbf{x}_0$ by colluding with other nodes.

Definition 3. Level-III Privacy: When the protocol ends, node 0 knows $\mathbf{x}_0$. A node i, where $i = 0, \ldots, m$, will not know $\mathbf{x}_j$ for $\forall j = 0, \ldots, m, j \neq i$. A node $i \neq 0$ cannot compute a coarse estimate of $\mathbf{x}_0$ even if it colludes with other nodes.

In all three levels of privacy, a node's coordinate is never disclosed to other nodes, for all anchors and the target. The main difference lies in the prevention of a coarse estimate about $\mathbf{x}_0$. With level-I privacy, an anchor will be able to compute by itself a coarse estimate of $\mathbf{x}_0$. Level-II privacy prevents an anchor from making such an estimation, but is vulnerable to collusion among anchors. However, note that even though collusion helps to estimate the location of the target, it does not help to compute the coordinates of other nodes. Finally, level-III privacy provides collusion-proof protection for both the actual target location and coarse estimate of the location.

C. Privacy Model

We assume that a participant of the localization, including both the anchors and the target, is honest but curious. A node executes the computation as specified by the protocol, but is curious about whatever information of others that could be leaked during the computation. In addition, we also assume that the communication between two nodes is encrypted, so that privacy leakage does not come from eavesdropping. We do not consider any active attack a node may launch, such as injection of false location information of the anchors, manipulation of the computation, or modification of (intermediate) results, with a purpose of misleading or cheating the target. All the above are valid attacks to the localization, but is out of the scope of this paper. Here we mainly focus on preventing privacy leakage in a normal localization computation.

Two scenarios will be considered in our privacy analysis: independent nodes and colluding nodes. For the former, information exchange between nodes only includes those specified by the protocol. As a result, a node can learn others' privacy only based on the legal information it receives. In contrast, for the latter scenario, colluding nodes may establish a side channel to exchange their information so as to figure out more information about others. In particular, colluding anchors can calculate a coarse estimate on $\mathbf{x}_0$ by pooling their location and ranging results together, so as to form a linear MMSE system similar to that of (6), but at a smaller scale. Moreover, our analysis also considers the scenario that the target colludes with some anchors to compute the location of other anchors.

D. Cryptographic Tool: Paillier Cryptosystem

Part of our constructions rely on the famous Paillier cryptosystem [19], a homomorphic encryption scheme that allows one to obtain the cipher text of an algebraic operation from the algebraic operation of the cipher text of the operands. Paillier cryptosystem is summarized below to facilitate the understanding of our protocols.

- Key generation: An entity chooses two primes p and q and compute $N = pq$ and $\lambda = \text{lcm}(p-1, q-1)$. It then selects a random $g \in \mathbb{Z}_{N^2}^*$ such that $\gcd(L(g^\lambda mod N^2), N) = 1$, where $L(x) = (x-1)/N$. The entity's Paillier public and private keys are $< N, g >$ and λ, respectively.

- Encryption: let $m \in \mathbb{Z}_N$ be a plaintext and $r \in F_N$ be a random number. The ciphertext is given by $E(m,r) = g^m r^N \mod N^2$.
- Decryption: Given a ciphertext $c \in \mathbb{Z}_{N^2}$, the plaintext is obtained by $D(c) = \frac{L(c^\lambda \mod N^2)}{L(g^\lambda \mod N^2)} \mod N$.

The Paillier cryptosystem has the following useful homomorphic property. For any $m_1, m_2, r_1, r_2 \in \mathbb{Z}_N$, we have

$$E(m_1,r_1)E(m_2,r_2) = E(m_1+m_2, r_1r_2) \mod N^2$$
$$E^{m_2}(m_1,r_1) = E(m_1m_2, r_1^{m_2}) \mod N^2.$$

We assume that N and g are 1024 and 160 bits, respectively, for sufficient semantical security [19]. Under this assumption, a Paillier encryption needs two 1024-bit exponentiations and one 2048-bit multiplication, and a Paillier decryption needs one 2048-bit exponentiation.

III. Privacy-Preserving Localization Protocols

We develop privacy-preserving localization protocols under the three aforementioned privacy levels, respectively. Among them, Protocol 1 has the lowest computation/communication overhead, but requires mobile anchors. Protocols 2 and 3 do not have this requirement and are applicable to both static and mobile anchors.

A. Protocol 1 for Level-I Privacy

Protocol 1 considers localization as an application of linear regression, and is based on the condition that an anchor is allowed to perform multiple ranging at different locations in localizing a target (so mobile anchor is assumed). The multilateration is based on the multiple ranging results of all the mobile anchors. Without loss of generality, suppose a node i, $i = 1, \ldots, m$, performs K ranging at K different locations, say $\mathbf{x}_i^{(1)}, \ldots, \mathbf{x}_i^{(K)}$, respectively (this can be easily extended to the case that node i performs ranging at K_i locations). Denote the result of the kth ranging be $d_{0i}^{(k)}$, where $k = 1, \ldots, K$. Following a similar linearization process to that in Section II-B, but this time the cancelation of the quadratic term is conducted between equations of the same anchor, a linear system describing the multi-lateration is obtained as follows: $\mathbf{R}\mathbf{x}_0^T = \mathbf{s}$, where

$$\mathbf{R} \overset{\text{def}}{=} 2\begin{bmatrix} x_{11}^{(K)} - x_{11}^{(1)} & \ldots & x_{1n}^{(K)} - x_{1n}^{(1)} \\ \vdots & \ddots & \vdots \\ x_{11}^{(K)} - x_{11}^{(K-1)} & \ldots & x_{1n}^{(K)} - x_{1n}^{(K-1)} \\ \vdots & \ddots & \vdots \\ x_{m1}^{(K)} - x_{m1}^{(1)} & \ldots & x_{mn}^{(K)} - x_{mn}^{(1)} \\ \vdots & \ddots & \vdots \\ x_{m1}^{(K)} - x_{m1}^{(K-1)} & \ldots & x_{mn}^{(K)} - x_{mn}^{(K-1)} \end{bmatrix} \tag{7}$$

$$\mathbf{s} \overset{\text{def}}{=} \begin{bmatrix} \sum_{j=1}^{n}(x_{1j}^{(K)^2} - x_{1j}^{(1)^2}) - (d_{01}^{(K)^2} - d_{01}^{(1)^2}) \\ \vdots \\ \sum_{j=1}^{n}(x_{1j}^{(K)^2} - x_{1j}^{(K-1)^2}) - (d_{01}^{(K)^2} - d_{01}^{(K-1)^2}) \\ \vdots \\ \sum_{j=1}^{n}(x_{mj}^{(K)^2} - x_{mj}^{(1)^2}) - (d_{0m}^{(K)^2} - d_{0m}^{(1)^2}) \\ \vdots \\ \sum_{j=1}^{n}(x_{mj}^{(K)^2} - x_{mj}^{(K-1)^2}) - (d_{0m}^{(K)^2} - d_{0m}^{(K-1)^2}) \end{bmatrix}. \tag{8}$$

The MMSE estimate for this system is calculated as $\mathbf{x}_0^T = (\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T\mathbf{s}$. To calculate $\mathbf{x}_0$ in a privacy-preserving fashion, each node follows the following protocol:

Protocol 1:

1) Anchor i, $i = 1, \ldots, m$, calculates $\Theta_i \overset{\text{def}}{=} \mathbf{R}_i^T\mathbf{R}_i$, and $\phi_i \overset{\text{def}}{=} \mathbf{R}_i^T\mathbf{s}_i$, where

$$\mathbf{R}_i \overset{\text{def}}{=} 2\begin{bmatrix} x_{i1}^{(K)} - x_{i1}^{(1)} & \ldots & x_{in}^{(K)} - x_{in}^{(1)} \\ x_{i1}^{(K)} - x_{i1}^{(2)} & \ldots & x_{in}^{(K)} - x_{in}^{(2)} \\ \vdots & \ddots & \vdots \\ x_{i1}^{(K)} - x_{i1}^{(K-1)} & \ldots & x_{in}^{(K)} - x_{in}^{(K-1)} \end{bmatrix}. \tag{9}$$

$$\mathbf{s}_i \overset{\text{def}}{=} \begin{bmatrix} \sum_{j=1}^{n}(x_{ij}^{(K)^2} - x_{ij}^{(1)^2}) - (d_{0i}^{(K)^2} - d_{0i}^{(1)^2}) \\ \sum_{j=1}^{n}(x_{ij}^{(K)^2} - x_{ij}^{(2)^2}) - (d_{0i}^{(K)^2} - d_{0i}^{(2)^2}) \\ \vdots \\ \sum_{j=1}^{n}(x_{ij}^{(K)^2} - x_{ij}^{(K-1)^2}) - (d_{0i}^{(K)^2} - d_{0i}^{(K-1)^2}) \end{bmatrix} \tag{10}$$

2) All anchors ($i = 1, \ldots, m$) send their Θ_i's and ϕ_i's to node 0. Node 0 calculates $\Theta \overset{\text{def}}{=} \sum_{i=1}^m \Theta_i$, $\phi \overset{\text{def}}{=} \sum_{i=1}^m \phi_i$, and computes $\mathbf{x}_0^T = \Theta^{-1}\phi$.

Theorem 1: Protocol 1 correctly calculates the MMSE estimate $\mathbf{x}_0$ for the linear system defined by (7) and (8).

Proof: Note that $\mathbf{R}$ and $\mathbf{s}$ defined in (7) and (8) can be written in terms of $\mathbf{R}_i$'s and $\mathbf{s}_i$'s as $\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_m \end{bmatrix}$ and $\mathbf{s} = \begin{bmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_m \end{bmatrix}$. Therefore, $\mathbf{R}^T\mathbf{R} = [\mathbf{R}_1^T \ldots \mathbf{R}_m^T]\begin{bmatrix} \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_m \end{bmatrix} = \sum_{i=1}^m \mathbf{R}_i^T\mathbf{R}_i = \sum_{i=1}^m \Theta_i = \Theta$. Similarly, $\mathbf{R}^T\mathbf{s} = [\mathbf{R}_1^T \ldots \mathbf{R}_m^T]\begin{bmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_m \end{bmatrix} = \sum_{i=1}^m \mathbf{R}_i^T\mathbf{s}_i = \sum_{i=1}^m \phi_i = \phi$. Therefore, $\mathbf{x}_0^T = (\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T\mathbf{s} = \Theta^{-1}\phi$. This proves Theorem 1. ■

Theorem 2: For independent nodes, Protocol 1 achieves Level-I privacy when $K > n + 1$.

Proof: The part related to coarse estimation of $\mathbf{x}_0$ is straightforward: Because an anchor i has K independent ranging results, it can use them to roughly estimate $\mathbf{x}_0$ as $(\mathbf{R}_i^T\mathbf{R}_i)^{-1}\mathbf{R}_i^T\mathbf{s}_i$. Next, we need to show that (1) an anchor j cannot compute another anchor's location $(\mathbf{x}_i^{(1)}, \ldots, \mathbf{x}_i^{(K)})$, for $i \neq j$; (2) an anchor cannot calculate node 0's MMSE location $\mathbf{x}_0$; and (3) node 0 cannot calculate any anchor's any location. The proof is given in [22] and is omitted here due to space limit. ■

Theorem 3: When there are node collusion, Protocol 1 achieves Level-I privacy when $K > n + 1$.

Proof: There are two possibilities for node collusion: (1) some anchors collude, or (2) some anchors collude with the target. For case (1), we can simply consider the colluded anchors as one virtual node. This essentially equals to a system with independent nodes. According to Theorem 2, Protocol 1 achieves Level-I privacy when $K > n + 1$. Similarly, for case (2), the collusion between the target and an anchor j will allow anchor j to learn other anchor, say anchor i's Θ_i and ϕ_i. But with this information, anchor j will not be able to figure out anchor i's location, otherwise node 0 would have already figured them out.

So such collusion does not increase the number of independent equations that can be used to solve the locations of those non-colluding anchors. Consequently, we may consider the colluded target and anchors as one virtual target. This essentially equals to a system with independent nodes. According to Theorem 2, Protocol 1 achieves Level-I privacy when $K > n+1$. This proves Theorem 3. ■

The computation overhead of Protocol 1 is dominated by the matrix multiplications at each anchor, which include one $n \times K-1$ matrix times one $K-1 \times n$ matrix, and one $n \times K-1$ matrix times one $K-1 \times 1$ vector. This amounts to roughly n^2K+nK multiplications per anchor, or $m[n^2K+nK]$ multiplications for all anchors. The communication overhead is due to the transmission of Θ_i and ϕ_i from anchor i, $i = 1, \ldots, m$, to node 0. This amounts to the communication of n^2+n real number per anchor, or $m(n^2+n)$ real number for all anchors.

B. Protocol 2 for Level-II Privacy

The essential reason that an anchor can obtain a coarse estimation on $\mathbf{x}_0$ in Protocol 1 is because the anchor is allowed to do ranging at multiple locations. This privacy leakage can be fixed by enforcing one ranging per anchor. This could be done, e.g., by all anchors measuring a pilot signal broadcasted by node 0, and node 0 only broadcasts this signal once. In this case, an anchor could be either static or mobile. The linear system describing the multi-lateration is defined by (4) and (5), and the MMSE estimate of $\mathbf{x}_0$ is given by (6).

The secure linear regression method used by Protocol 1 is no longer privacy-preserving when being used to compute (6). To see this, similar to the definition of $\mathbf{R}_i$ and $\mathbf{s}_i$ in (9) and (10), now for nodes $i = 1, \ldots, m-1$, we define $\mathbf{A}_i \stackrel{\text{def}}{=} 2\begin{bmatrix} x_{m1}-x_{i1} & x_{m2}-x_{i2} & \ldots & x_{mn}-x_{in}\end{bmatrix}$ and $b_i \stackrel{\text{def}}{=} \sum_{j=1}^{n}(x_{mj}^2 - x_{ij}^2) - (d_{0m}^2 - d_{0i}^2)$ (instead of a vector, b_i degenerates to a scalar). Two intermediate steps in the linear regression leak privacy between nodes: (1) In order for node i to construct $\mathbf{A}_i$ and b_i, it requires node m to disclose $\mathbf{x}_m$ and d_{0m} to every other anchor; (2) When anchor i sends $\Theta_i = \mathbf{A}_i^T\mathbf{A}_i$ and $\phi_i = \mathbf{A}_i^T b_i$ to node 0, node 0 can computes the elements in $\mathbf{A}_i$ and b_i from Θ_i and ϕ_i. Therefore, collectively, node 0 can recover $\mathbf{A}$ and $\mathbf{b}$, and thus the location of every anchor, from Θ_i's and b_i's, $i = 1, \ldots, m-1$. To enable privacy-preserving localization in this case, Protocol 2 is developed below.

We rewrite $\mathbf{A}$ as follows:

$$\mathbf{A} = \sum_{i=1}^{m}\mathbf{M}_i \tag{11}$$

where $\mathbf{M}_i$ is a $(m-1)\times n$ matrix defined as

$$\mathbf{M}_i \stackrel{\text{def}}{=} \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ -x_{i1} & -x_{i2} & \cdots & -x_{in} \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad \text{for } i = 1, \ldots, m-1 \tag{12}$$

where all rows other than the ith row are 0. $\mathbf{M}_m$ is a $(m-1)\times n$ matrix defined as

$$\mathbf{M}_m \stackrel{\text{def}}{=} \begin{bmatrix} x_{m1} & x_{m2} & \cdots & x_{mn} \\ x_{m1} & x_{m2} & \cdots & x_{mn} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \tag{13}$$

Note that anchor i is able to construct $\mathbf{M}_i$, for $i = 1, \ldots, M$, based on its own knowledge. Because the row vector $\mathbf{x}_i = (x_{i1} \ldots x_{in})$, the above can be concisely written as $\mathbf{M}_i = \begin{bmatrix} 0 \\ \vdots \\ -\mathbf{x}_i \\ \vdots \\ 0 \end{bmatrix}$ for $i = 1, \ldots, m-1$, and $\mathbf{M}_m = \begin{bmatrix} \mathbf{x}_m \\ \vdots \\ \mathbf{x}_m \end{bmatrix}$. Accordingly,

$$\begin{aligned} \mathbf{A}^T\mathbf{A} &= (\sum_{i=1}^{m}\mathbf{M}_i)^T(\sum_{j=1}^{m}\mathbf{M}_j) \\ &= \sum_{i=1}^{m}\sum_{j=1}^{m}\mathbf{M}_i^T\mathbf{M}_j \end{aligned} \tag{14}$$

It is easy to show that

$$\mathbf{M}_i^T\mathbf{M}_j = \begin{cases} 0, & \text{when } i \neq j \text{ and } i,j \neq M \\ \mathbf{x}_i^T\mathbf{x}_i, & \text{when } i = j \text{ and } i,j \neq M \\ -\mathbf{x}_i^T\mathbf{x}_m, & \text{when } i \neq m \text{ and } j = m \\ -\mathbf{x}_m^T\mathbf{x}_j, & \text{when } i = m \text{ and } j \neq m \\ (m-1)\mathbf{x}_m^T\mathbf{x}_m, & \text{when } i = j = m \end{cases} \tag{15}$$

Therefore

$$\mathbf{A}^T\mathbf{A} = (m-1)\mathbf{x}_m^T\mathbf{x}_m + \sum_{i=1}^{m-1}\mathbf{x}_i^T\mathbf{x}_i - (\sum_{i=1}^{m-1}\mathbf{x}_i^T)\mathbf{x}_m - \mathbf{x}_m^T(\sum_{i=1}^{m-1}\mathbf{x}_i) \tag{16}$$

Similarly, $\mathbf{b}$ in (5) can be rewritten as $\mathbf{b} = \sum_{i=1}^{m}\mathbf{h}_i$, where $\mathbf{h}_i$ is a $(m-1)\times 1$ column vector defined as

$$\mathbf{h}_i \stackrel{\text{def}}{=} \begin{bmatrix} 0 \\ \vdots \\ -\sum_{j=1}^{n}x_{ij}^2 + d_{0i}^2 \\ \vdots \\ 0 \end{bmatrix} \quad \text{for } i = 1, \ldots, m-1 \tag{17}$$

where all elements other than the ith row are 0. $\mathbf{h}_m$ is a $(m-1)\times 1$ column vector defined as

$$\mathbf{h}_m \stackrel{\text{def}}{=} \begin{bmatrix} \sum_{j=1}^{n}x_{mj}^2 - d_{0m}^2 \\ \vdots \\ \sum_{j=1}^{n}x_{mj}^2 - d_{0m}^2 \end{bmatrix} \tag{18}$$

Therefore,

$$\mathbf{A}^T\mathbf{b} = \sum_{i=1}^{m}\sum_{j=1}^{m}\mathbf{M}_i^T\mathbf{h}_j \tag{19}$$

Defining $h_i \stackrel{\text{def}}{=} \sum_{j=1}^{n}x_{ij}^2 - d_{0i}^2$ for $i = 1, \ldots, m$, it can be shown that

$$\mathbf{M}_i^T\mathbf{h}_j = \begin{cases} 0, & \text{when } i \neq j \text{ and } i,j \neq m \\ h_i\mathbf{x}_i^T, & \text{when } i = j \text{ and } i,j \neq m \\ -h_m\mathbf{x}_i^T, & \text{when } i \neq m \text{ and } j = m \\ -h_j\mathbf{x}_m^T, & \text{when } i = m \text{ and } j \neq m \\ (m-1)h_m\mathbf{x}_m^T & \text{when } i = j = m \end{cases} \tag{20}$$

Therefore

$$\mathbf{A}^T\mathbf{b} = (m-1)h_m\mathbf{x}_m^T + \sum_{i=1}^{m-1}h_i\mathbf{x}_i^T - h_m(\sum_{i=1}^{m-1}\mathbf{x}_i^T) - (\sum_{i=1}^{m-1}h_i)\mathbf{x}_m^T \tag{21}$$

An observation on (16) and (21) shows that in these equations, the first two terms can be calculated by anchor m and anchors $i = 1, \ldots, m-1$, respectively, based on their own knowledge, and the last two terms are based on anchor m and the aggregation of anchors 1 through $m-1$. Based on this observation, Protocol 2 obtains privacy-preserving localization as follows:

Protocol 2:

1) Every node $i = 1, \ldots, m$ generates m random $n \times n$ matrices $\mathbf{p}_i^{(k)}$, where $k = 1, \ldots, m$, such that $\sum_{k=1}^{m} \mathbf{p}_i^{(k)} = 0$. Node i keeps one such matrix, and sends the rest to the other $m-1$ nodes, respectively. Node i creates $\mathbf{P}_i$ by adding up all $m-1$ matrices it receives from other $m-1$ nodes, with the one it keeps. Note that $\mathbf{P}_i$ is a random matrix, and $\sum_{i=1}^{m} \mathbf{P}_i = 0$.
2) In a similar way to Step 1, node $i = 1, \ldots, m$ generates random $n \times 1$ vector $\mathbf{v}_i$, such that $\sum_{i=1}^{m} \mathbf{v}_i = 0$.
3) In a similar way to Step 1, but this time applied only to nodes $i = 1, \ldots, m-1$, anchor i generates a random $n \times 1$ vector $\mathbf{w}_i$, such that $\sum_{i=1}^{m-1} \mathbf{w}_i = 0$.
4) In a similar way to Step 1, anchor i, where $i = 1, \ldots, m-1$, generates a random number t_i, such that $\sum_{i=1}^{m-1} t_i = 0$.
5) Anchor i, $i = 1, \ldots, m-1$, calculates and sends $\Omega_i \overset{\text{def}}{=} \mathbf{x}_i^T \mathbf{x}_i + \mathbf{P}_i$ and $\psi_i \overset{\text{def}}{=} h_i \mathbf{x}_i^T + \mathbf{v}_i$ to the target, and calculates and sends $\alpha_i \overset{\text{def}}{=} \mathbf{x}_i^T + \mathbf{w}_i$ and $\beta_i \overset{\text{def}}{=} h_i + t_i$ to node m.
6) Node m calculates $\alpha = \sum_{i=1}^{m-1} \alpha_i$ and $\beta = \sum_{i=1}^{m-1} \beta_i$. It then calculates and sends $\Omega_m \overset{\text{def}}{=} (m-1)\mathbf{x}_m^T \mathbf{x}_m - \alpha \mathbf{x}_m - \mathbf{x}_m^T \alpha^T + \mathbf{P}_m$ and $\psi_m \overset{\text{def}}{=} (m-1) h_m \mathbf{x}_m^T - h_m \alpha - \beta \mathbf{x}_m^T + \mathbf{v}_m$ to the target.
7) Node 0 calculates $\Omega \overset{\text{def}}{=} \sum_{i=1}^{m} \Omega_i$ and $\psi \overset{\text{def}}{=} \sum_{i=1}^{m} \psi_i$. It then calculates $\mathbf{x}_0^T = \Omega^{-1} \psi$.

Theorem 4: Protocol 2 correctly calculates the MMSE estimate $\mathbf{x}_0$ for the linear system defined in (4) and (5).

Proof: The proof is straightforward based on the discussion before the Theorem, and therefore is omitted here due to space limit. ■

Theorem 5: For independent nodes, Protocols 2 achieves Level-II privacy when $m > n$, where m is the number of anchors and n is the dimensionality of the physical space to perform localization.

Proof: 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 compute a coarse estimate about the location of the target, and (3) the target cannot learn the location of any anchor. The full proof is given in [22] and is omitted here due to space limit. ■

Theorem 6: When the number of colluding anchors is less than half of $m-1$ and the number of non-colluding anchors is greater than $n+1$, Protocol 2 achieves Level-II privacy.

Proof: When anchors collude, the leak of a coarse estimate of $\mathbf{x}_0$ by protocol 2 is inevitable, because the colluding anchors can pool their location and ranging information together to construct a smaller-scale multi-lateration linear system to locate the target. To prove the rest of the Theorem, we need to show that the collusion does not help to reveal the location of the target and non-colluding anchors. We consider the following two collusion scenarios: (1) the colluded nodes do not include the target and (2) the colluding nodes include the target. The full proof is given in [22] and is omitted here due to space limit. ■

The computation overhead of Protocol 2 is dominated by the vector multiplications in Steps 5 and 6. Specifically, a node i, $i = 1, \ldots, m-1$, needs to perform $n^2 + n$ multiplications in Step 5. Node m performs roughly $3n^2 + 4n$ multiplications in Step 6. So, the total number of multiplications for all m anchors is $n^2(m+2) + n(m+3)$. For a node i, $1 \leq i \leq m-1$, the numbers of elements it transmits in Protocol 2 are $n^2(m-1)$ (in Step 1), $n(m-1)$ (in Step 2), $n(m-2)$ (in Step 3), $m-2$ (in Step 4), n^2+2n+1 (in Step 5), or $n^2m + n(2m-1) + m - 1$ per anchor. For node m, the numbers of elements it transmits in the Protocol are $n^2(m-1)$ (in Step 1), $n(m-1)$ (in Step 2), and $n^2 + n$ (in Step 6), or $n^2m + nm$ all steps together. So the total number of elements transmitted in the protocol is roughly $n^2m^2 + (2n+1)m^2$. Assuming each element is represented by 24 bits, in total Protocol 2 needs to transmit $[n^2m^2 + (2n+1)m^2] \times 24$ bits in one localization operation.

C. Protocol 3 for Level-III Privacy

In Protocol 2, the main reason that a group of colluding anchors can calculate a coarse estimate of $\mathbf{x}_0$ is because an anchor has the knowledge of both its location and the ranging information. This privacy breach can be fixed by separating the ownership of these two information. In particular, an anchor still knows its location, but the target will be the one to perform ranging, for every anchor. Anchors are required to transmit a pilot signal by turns, and the target estimates the distance to every anchor via the received signal strength of that anchor's pilot signal. As a result, d_{0i}, $i = 1, \ldots, m$, becomes the private information of the target. So the privacy-preserving localization problem becomes how to compute (6) based on the private ranging information of the target and the private location information of the anchors.

Our solution is built upon Protocol 3, with a modified component that calculates the cross terms between d_{0i}'s and $\mathbf{x}_i$'s in a privacy-preserving fashion. More specifically, it can be observed that the above cross terms only appear in the calculation of $\mathbf{A}^T\mathbf{b}$ (i.e., (19)). To separate the calculation of the cross terms, (19) can be rewritten as follows

$$\mathbf{A}^T\mathbf{b} = \sum_{i=1}^{m} \sum_{j=1}^{m} \mathbf{M}_i^T \mathbf{h}'_j + \sum_{i=1}^{m} \mathbf{M}_i^T \mathbf{d} \tag{22}$$

where $\mathbf{h}'_i \overset{\text{def}}{=} \begin{bmatrix} 0 \\ \vdots \\ -\sum_{j=1}^{n} x_{ij}^2 \\ \vdots \\ 0 \end{bmatrix}$, for $i = 1, \ldots, m-1$, $\mathbf{h}'_m \overset{\text{def}}{=} \begin{bmatrix} \sum_{j=1}^{n} x_{mj}^2 - d_{0m}^2 \\ \vdots \\ \sum_{j=1}^{n} x_{mj}^2 - d_{0m}^2 \end{bmatrix}$, and $\mathbf{d} \overset{\text{def}}{=} \begin{bmatrix} d_{01}^2 - d_{0m}^2 \\ d_{02}^2 - d_{0m}^2 \\ \vdots \\ d_{0m-1}^2 - d_{0m}^2 \end{bmatrix}$.

Modifying the definition of h_i in Protocol 2 to $h_i \overset{\text{def}}{=} \sum_{j=1}^{n} x_{ij}^2$, for $i = 1, \ldots, m$, it is clear that the first term on the RHS of (22) can be securely computed using Protocol 2. As a result, $\sum_{i=1}^{m} \sum_{j=1}^{m} \mathbf{M}_i^T \mathbf{h}'_j = \psi$, where ψ is calculated according to Step 7 of Protocol 2.

Defining $d_i \overset{\text{def}}{=} d_{0i}^2 - d_{0m}^2$, $\mathbf{M}_i^T\mathbf{d}$ can be calculated as $\mathbf{M}_i^T\mathbf{d} = \begin{bmatrix} -x_{i1}d_i \\ -x_{i2}d_i \\ \vdots \\ -x_{in}d_i \end{bmatrix}$, for $i = 1, \ldots, m-1$, and $\mathbf{M}_m^T\mathbf{d} = \begin{bmatrix} x_{m1}d_\Sigma \\ x_{m2}d_\Sigma \\ \vdots \\ x_{mn}d_\Sigma \end{bmatrix}$, where $d_\Sigma \overset{\text{def}}{=} \sum_{j=1}^{m-1} d_i$. So the second term on the RHS of (22),

$\sum_{i=1}^{m} \mathbf{M}_i^T\mathbf{d}$, can be securely computed using the following Pailliar homomorphic encryption algorithm.

Algorithm 1:

1) Every node $i = 1, \ldots, m$ generates m random $n \times 1$ vectors $\mathbf{z}_i^{(k)}$, where $k = 1, \ldots, m$, such that $\sum_{k=1}^{m} \mathbf{z}_i^{(k)} = 0$. Node i keeps one such vector, and sends the rest to the other $m-1$ nodes, respectively. Node i creates vector $\mathbf{Z}_i$ by adding up all $m-1$ vectors it receives from other $m-1$ nodes, with the one it keeps. As a result, $\mathbf{Z}_i$ is a random vector, and $\sum_{i=1}^{m} \mathbf{Z}_i$.
2) For node $i = 1, \ldots, m-1$, the target securely calculates $\mathbf{M}_i^T\mathbf{d} + \mathbf{Z}_i$ in the following way
   a) Using its public Paillier key, the target calculates the following ciphertexts for node i: $E_0(-d_i)$ and $E_0(1)$, and sends these ciphertexts to node i.
   b) Node i calculates the following sequentially for $j = 1, \ldots, n$: $E_0^{x_{ij}}(-d_i) = E_0(-x_{ij}d_i)$, $E_0^{Z_{ij}}(1) = E_0(Z_{ij})$, $E_0(-x_{ij}d_i)E_0(Z_{ij}) = E_0(-x_{ij}d_i + Z_{ij})$, where Z_{ij} is the jth element of vector $\mathbf{Z}_i$. Node i sends $E_0(-x_{ij}d_i + Z_{ij})$, $j = 1 \ldots n$, to the target.
   c) The target decrypts $E_0(-x_{ij}d_i + Z_{ij})$, $j = 1 \ldots n$, to construct $\mathbf{M}_i^T\mathbf{d} + \mathbf{Z}_i$.
3) For node m, the target securely computes $\mathbf{M}_m^T\mathbf{d} + \mathbf{Z}_m$ in the following way
   a) Using its public Paillier key, the target calculates the following ciphertexts, $E_0(d_\Sigma)$ and $E_o(1)$, and sends these ciphertexts to node m.
   b) Node m calculates the following sequentially for $j = 1 \ldots n$: $E_0^{x_{mj}}(d_\Sigma) = E_0(x_{mj}d_\Sigma)$, $E_0^{Z_{mj}}(1) = E_0(Z_{mj})$, $E_0(x_{mj}d_\Sigma)E_0(Z_{mj}) = E_0(x_{mj}d_\Sigma + Z_{mj})$. Node m sends $E_0(x_{mj}d_\Sigma + Z_{mj})$, $j = 1 \ldots n$, to the target.
   c) The target decrypts $E_0(x_{mj}d_\Sigma + Z_{mj})$, $j = 1 \ldots n$, to construct $\mathbf{M}_m^T\mathbf{d} + \mathbf{Z}_m$.
4) The target calculates $\psi' \overset{\text{def}}{=} \sum_{i=1}^{m} \mathbf{M}_i^T\mathbf{d} = \sum_{i=1}^{m}(\mathbf{M}_i^T\mathbf{d} + \mathbf{Z}_i)$

Based on the above algorithms, collusion-resilient Protocol 3 is as follows:

Protocol 3

1) Based on the revised definition of $h_i = \sum_{j=1}^{n} x_{ij}^2$, execute Protocol 2. The target node obtains Ω and ψ.
2) Execute Algorithm 1. The target obtains ψ'.
3) The target calculates $\mathbf{x}_0^T = \Omega^{-1}(\psi + \psi')$.

Theorem 7: Protocol 3 correctly calculates the MMSE estimate $\mathbf{x}_0$ for the linear system defined in (4) and (5).

Proof: The proof is straightforward based on the discussion preceding the protocol, and therefore is omitted here due to space limit. ■

Theorem 8: For both independent and colluding-node cases, as long as the number of colluding nodes is less than half of $m-1$ and the number of non-colluding nodes is greater than $n+1$, Protocol 3 achieves Level-III privacy.

Proof: Because Protocol 3 is built upon Protocol 2, and we have proved that Protocol 2 achieves Level-II privacy, here we only need to show that (1) under Protocol 3 an anchor cannot compute a coarse estimate about $\mathbf{x}_0$, no matter it colludes with other anchors or not, (2) a node cannot compute any other node's location, no matter it colludes with other nodes or not. The full proof is given in [22] and is omitted here due to space limit. ■

The computation overhead of Protocol 3 is dominated by the secure computation of the Paillier cryptosystem in Steps 2 and 3 of Algorithm 1. For every node i, $i = 1, \ldots, m$, the calculation in Steps 2 and 3 of Algorithm 1 involves one Paillier encryption, one Paillier decryption, $2n$ 2048-bit exponentiations, and n 2048-bit multiplications. Overall, this amounts to $2m$ 1024-bit exponentiations, $(n+1)m$ 2048-bit multiplications, and $(2n+1)m$ 2048-bit exponentiations for all the nodes per localization operation.

The communication overhead of Algorithm 1 is mainly due to the Paillier secure computation in Steps 2 and 3, and the exchange of vectors $\mathbf{z}_i^{(k)}$ in Step 1. In particular, for a node i, $i = 1, \ldots, m$, it transmits $m-1$ $n \times 1$ real vectors in Step 1, and transmits $2048 \times n$ bits ciphertext of the secure computation results in Steps 2 or 3. Assuming that an element of the vector $\mathbf{z}_i^{(k)}$ is represented by 24 bits, the total traffic transmitted in Algorithm 1 is $2048mn + 24m(m-1)n$ bits. Therefore, in total Protocol 3 needs to transmit roughly $2048mn + 24[n^2m^2 + (3n+1)m^2]$ bits in one localization operation.

IV. Performance Evaluation

A. Evaluation Setup

In this section, we compare the computation and communication overhead of the proposed protocols with prior results based on numerical examples. We are not aware of any existing algorithm that is specifically designed for preserving the multi-lateral privacy in localization. Therefore, we only consider the general multi-party secure LSE algorithm that can be applied to compute the MMSE estimate of $\mathbf{x}_0$ (i.e., equation (6)) in a privacy-preserving fashion. In particular, we consider two well-known algorithms: one based on oblivious transfer (OT) [8] and the other on homomorphic encryption (HE) [12]. The original design of both algorithms only considers secure computation between two parties. A straightforward extension to m-party ($m > 2$) secure computation requires executing the 2-party algorithm for every pair of nodes [12]. Therefore, the computation and communication overhead of the m-party computation is m^2 times of that of the 2-party one. Moreover, note that OT and HE cannot prevent anchors from guessing $\mathbf{x}_0$ by forming collusion, and therefore they can only achieve Level-II privacy. The level of privacy, computation complexity, and communication cost (in number of transmitted bits) of the proposed protocols and the prior algorithms are summarized in Table I.

In Table I, μ is the protection parameter for the oblivious transfer operation in OT. As suggested by [8], $\mu = 256$ is assumed. We also have assumed that a real number is represented by 24 bits. The notations of χ_1, χ_2, ε_1, and ε_2 represent the operations of 24-bit multiplication, 2048-bit multiplication, 1024-bit exponentiation, and 2048-bit exponentiation, respectively. In our numerical examples, we assume the following execution time for these operations: $\chi_1 = 1$ μs, $\chi_2 = 0.88$ ms, $\varepsilon_1 = 81.08$ ms, and $\varepsilon_2 = 159.06$ ms. The setting of these parameters is based on the mean value of the benchmark test result in [29], which is obtained on a LG P-970 smartphone equipped with a 1 GHz Cortex-A8 CPU, 512 MB RAM, and Android v2.2 OS. We also assume communication between nodes has a bandwidth of 2 Mbps.

Our performance metrics include total computation time, total number of transmitted bits, and the protocol execution time. The first two metrics measure the summation of the CPU time and the numbers of bits transmitted over all participants

Algorithm	Privacy	Computation	Communication
Protocol 1	Level-I	$m(n^2K + nK)\chi_1$	$m(n^2 + n) \times 24$
Protocol 2	Level-II	$[n^2(m+2) + n(m+3)]\chi_1$	$[n^2m^2 + (2n+1)m^2] \times 24$
Protocol 3	Level-III	$2m\varepsilon_1 + (n+1)m\chi_2 + (2n+1)m\varepsilon_2$	$2048mn + 24[n^2m^2 + (3n+1)m^2]$
Homomorphic Encryption	Level-II	$2m^3n^2\chi_2 + m^2(m+1)n^2\varepsilon_2 + 2m^3n^2\varepsilon_1$	$2048(m^3n + m^2n^2)$
Oblivious Transfer	Level-II	$\mu m^3(n^2 + n)\chi_1$	$\mu m^3 n \times 24$

TABLE I
PROTOCOL OVERHEAD.

of the localization. The protocol execution time is defined as the summation of time consumed by each step of the protocol, including both computation and communication overhead. The steps that are executed in parallel by multiple nodes are only counted once. We only present the results for the 2-D localization ($n = 2$), due to its popularity. The trends for 3-D case is similar.

B. Numerical Results

We plot the computation cost as a function of the number of anchors in Figure 1. It can be observed that for protocols 1 through 3, their computation cost increases with their level of privacy. This is not surprising, as a higher privacy level implies more protection, which can only be obtained by more complicated computation. Moreover, the proposed protocols are much more computationally efficient than HE and OT. In particular, Protocols 1 and 2 reduce the total CPU time by at least 2 orders of magnitude when compared to HE and OT. The computation cost of Protocol 3 is about $1/10$ to $1/100$ to that of HE, and is comparable to that of OT. In general, protocols that are based on cryptographic encryptions are much more computationally expensive than the ones that are not, because of the long-bit multiplications and exponentiations required by the encryption/decryption. Protocol 3 has a lower computation cost than HE, because only part of its construction is based on Paillier encryption. In contrast, the HE algorithm fully relies on homomorphic encryption.

We compare the communication cost of various protocols in Figure 2. It can be observed that, for Protocols 1 through 3, their communication overhead increases with the level of privacy–a phenomenon similar to their computation cost. On the other hand, their communication cost is much smaller than that of HE and OT. HE has a high communication cost because its calculation is fully performed in the encrypted space. The input of the calculation, i.e., the ciphertext, has 2048 bits and is much longer than the 24-bit real number used in Protocols 1 through 2. On the other hand, Protocol 3 is only partially based on homomorphic encryption, and therefore requires less transmission of ciphertexts, yielding higher communication efficiency than HE. The high communication cost of OT is resulted from the large number of random matrices transmitted between each pair of nodes in the oblivious transfer operation.

The protocol execution time of various mechanisms is plotted as a function of the number of anchors in Figure 3. Once again, it can be observed that the execution time of the 3 proposed protocols increases with their privacy level, but are all smaller than that of the generic HE and OT algorithms. In particular, the execution time of Protocols 1 through 3 ranges from a few ms to hundreds of ms. This indicates that the proposed protocols are very practical. Moreover, it can be observed from Figure 3 that the execution time of Protocol 3 changes little with the number of anchors. This is not surprising, as the execution time of Protocol 3 is dominated by the Paillier-based secure computation of $\mathbf{M}_i^T\mathbf{d} + \mathbf{Z}_i$'s, which can be distributed to each anchor and be computed in parallel by all the anchors.

V. RELATED WORK

Despite the large body of work on privacy-preserving access to LBS, only few studies address privacy-preserving localization. Existing work mainly focuses on protecting the unilateral privacy of the target using physical layer technologies. Based on the basic observation that an adversary needs to be within the communication range of the target in order to calculate its location, early work reduces the adversary's chance of attack by reducing the spatial footprint of the target's communication. This is achieved by either reducing the target's communication range through power control [13], or by changing the transmission from omnidirectional to a shaped beam using antenna arrays [26]. A side effect of these approaches is the reduced number of anchors in the target's communication range, and hence the localization accuracy is compromised. Subsequent methods overcome this weakness by optimizing the radiation pattern of the antenna array so that its location privacy is protected while the communication quality is not affected. In particular, [24] proposed methods of antenna pattern synthesis to create forged location. [18] extends the effort to multiple mobile nodes by leveraging cooperation among nodes in close vicinity and utilizing synchronized transmissions to obfuscate localization of adversary. Unilateral localization privacy is also achieved by the target intentionally injecting a measurement error, which is a secret held by the target, into the ranging outcome. As a result, the target is the only one that can remove the error and calculate the right location. [30] proposed to induce such a measurement error by manipulating the signal's propagation time. [2] achieves the same goal for a RFID system by controlling RFID tag's response time to reader's inquiry.

Different from the previous studies, we develop multi-lateral localization privacy preservation techniques to protect not only the target location but also the location information of the anchors together with any side information that could derive the coarse-grained position of the target. We formulate our problem as a secure least-squared-error (LSE) estimation for an over-determined linear system. Although a secure LSE problem can in general be solved using the classical secure multi-party computation (SMC) techniques [10], e.g., the secure computation circuit method [28], [10], the oblivious transfer method [8], and the method fully based on homomorphic encryption [12], it suffers from high computation/communication cost. To lower the cost, in practice these methods are typically used for two-party computations only, e.g., the secure calculation of set intersection [6], [15] or the privacy-preserving matching [7], [29]. In contrast, our problem involves computation among many parties in order to achieve high localization accuracy, and requires high computation/communication efficiency due to the severe resource constraints in mobile computing. This renders the existing methods not suitable to address the unique multi-lateral privacy preserving localization problem in mobile environments. Although efficient solutions have been proposed for secure LSE problem based on the commodity-server framework [9], [14], these solutions require a trusted central server in the computation. Such solutions are not applicable to our

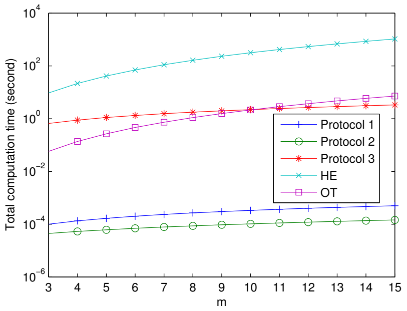

Fig. 1. Computation cost.

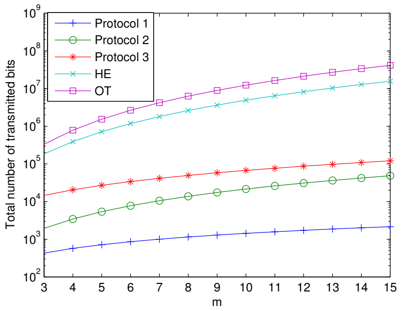

Fig. 2. Commun. cost.

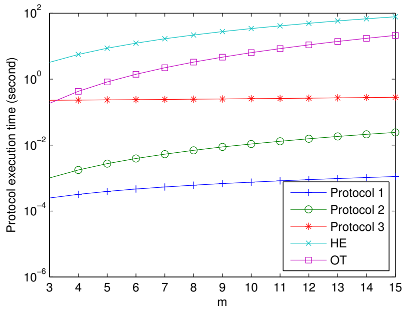

Fig. 3. Protocol exe time.

problem, because ours has a distributed setup and no trusted central server can be assumed.

On a different track, the problem of secure localization has been studied in the literature, which focus on detecting corrupted target's localization results and developing techniques to correct them. In particular, the verifiable multi-lateration mechanism is proposed in [3], which uses distance bounding protocols for secure position computation and verification. The followup work in [4] further proposes to use hidden and mobile base stations to localize and verify location estimates. Note that our study addresses a different problem by focusing on the privacy aspect of localization process.

VI. Conclusions

In this paper, we address the privacy leakage problem during the localization process, and prevent the leakage of the location information of both the target as well as anchors simultaneously. We have developed three multi-lateral privacy-preserving localization schemes that can provide different levels of protection for any intermediate location-related information and resilience to anchor node collusion, in addition to the unique capability of hiding the exact location for both the target and anchors at the same time. By taking advantage of the combinations of information hiding and homomorphic encryption, the proposed schemes only incur low cost in computation/communication overhead, and can trade user's privacy requirements for better computation/communication efficiency, which is especially desirable in a resource-constrained mobile computing environment. Our current constructions are based on the popular multi-lateration/triangulation localization models involving ranging. Our future work will extend privacy-preserving localization into range-free models, such as those based on signal fingerprints.

Acknowledgements

T. Shu is supported in part by NSF Award CNS-1343156; Y. Chen is supported by NSF Awards CNS-1318748 and CNS-0954020, and by Army Research Office W911NF-13-1-0288; J. Yang is supported in part by NSF Award CNS-1318751.

References

[1] A. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[2] M. Burmester. Localization privacy. *Cryptography and Security: From Theory to Applicatons, Lecture Notes in Computer Science*, 6805:425–441, 2012.

[3] S. Capkun and J. Hubaux. Secure positioning of wireless devices with application to sensor networks. In *IEEE INFOCOM*, 2005.

[4] S. Capkun and J. Hubaux. Securing localization with hidden and mobile base stations. In *IEEE INFOCOM*, 2006.

[5] Z. Chen, X. Hu, X. Ju, and K. Shin. LISA: location information scrambler for privacy protection on smartphones. In *IEEE CNS'13*, 2013.

[6] E. Cristofaro and G. Tsudik. Practical private set intersection protocols with linear complexity. In *Proceedings of FC'10*, volume 6052, Jan. 2010.

[7] W. Dong, V. Dave, L. Qiu, and Y. Zhang. Secure friend discovery in mobile social networks. In *IEEE INFOCOM*, Apr. 2011.

[8] W. Du and M. J. Atallah. Privacy-preserving cooperative scientific computations. In *Proceedings of 14th IEEE Computer Security Foundations Workshop*, pages 273–282, June 2001.

[9] W. Du and Z. Zhan. A practical approach to solve secure multi-party computation problems. In *Proceedings of New Security Paradigms Workshop*, pages 127–135, Sept. 2002.

[10] O. Goldreich. Secure multi-party computation. *working draft, available at http://www.wisdom.weizmann.ac.il/home/oded/public_html/foc.html*.

[11] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *ACM MOBISYS*, 2003.

[12] R. Hall, A. Rinaldo, and L. Wasserman. Secure multiparty linear regression based on homomorphic encryption. *Journal of Official Statistics*, 27(4):669–691, 2011.

[13] T. Jiang, H. J. Wang, and Y. C. Hu. Preserving location privacy in wireless LANs. In *ACM MOBISYS*, 2007.

[14] J. Kang and D. Hong. A practical privacy-preserving cooperative computation protocol without oblivious transfer for linear systems of equations. *International Journal of Information Processing Systems*, 3(1):21–25, 2007.

[15] L. Kissner and D. Song. Privacy-preserving set operations. In *Proceedings of CRYPTO'05*, Aug. 2005.

[16] X. Li, Y. Chen, J. Yang, and X. Zheng. Achieving robust wireless localization resilient to signal strength attacks. *Springer Wireless Networks (WiNET), the Journal of Mobile Communication, Computation and Information*, 18(1):45–58, 2012.

[17] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye. Push the limit of WiFi based localization for smartphones. In *ACM MobiCom*, Aug. 2012.

[18] S. Oh, T. Vu, M. Gruteser, and S. Banerjee. Phantom: physical layer cooperation for location privacy protection. In *IEEE INFOCOM*, 2012.

[19] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of EUROCRYPT'99*, May 1999.

[20] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: zero-effort crowdsourcing for indoor localization. In *ACM MobiCom*, pages 293–304, 2012.

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

[22] T. Shu, Y. Chen, J. Yang, and A. Williams. Multi-lateral privacy-preserving localization in pervasive environments. *Technical Report of Oakland University, Department of Computer Science and Engineering, available at http://www.secs.oakland.edu/∼shu/*, 2013.

[23] A. Uchiyama, S. Fujii, K. Maeda, T. Umedu, H. Yamaguchi, and T. Higashino. UPL: opportunistic localization in urban districts. *IEEE Transactions on Mobile Computing*, 12(5), May 2013.

[24] T. Wang and Y. Yang. Location privacy protection from RSS localization system using antenna pattern synthesis. In *IEEE INFOCOM*, 2011.

[25] T. Wang and Y. Yang. Analysis on perfect location spoofing attacks using beamforming. In *IEEE INFOCOM*, 2013.

[26] F. L. Wong, M. Lin, S. Nagaraja, I. Wassell, and F. Stajano. Evaluation framework of location privacy of wireless mobile systems with arbitrary beam pattern. In *CNSR'07*, 2007.

[27] J. Yang and Y. Chen. Towards attack resistant localization under infrastructure attacks. *Security and Communication Networks (SCN), Wiley*, 5(4):384–403, 2012.

[28] A. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, 1982.

[29] 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*, to appear.

[30] S. Zhong, L. Li, Y. Liu, and Y. R. Yang. Privacy-preserving location-based services for mobile users in wireless networks. *Yale Computer Science Technical Report YALEU/DCS/TR-1297, available at http://www.cs.yale.edu/research/techreports/tr1297.pdf*, 2004.