Protecting Multi-Lateral Localization Privacy in Pervasive Environments

Tao Shu, Yingying Chen, and Jie Yang, *Member, IEEE*

***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 a user's location in accessing the LBS server, the possible privacy leakage in the calculation of the user's location, i.e., the localization, has been largely ignored. Such a privacy leakage stems from the fact that a localization algorithm typically takes the location of anchors (reference points for localization) as input, and generates the target's location as output. As such, the location of anchors and target 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, i.e., the target's location, is strictly limited to the target itself. To fully protect the 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. We formulate the problem as a secure least-squared-error (LSE) estimation for an overdetermined linear system and develop three privacy-preserving solutions 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 a user's privacy requirements for better computation and communication efficiency. Through numerical results, we verify the significant efficiency improvement of the proposed schemes over existing multiparty secure LSE algorithms.**

***Index Terms*—Homomorphic encryption, localization, location privacy.**

Manuscript received September 08, 2014; revised April 20, 2015; accepted August 19, 2015; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor A. X. Liu. Date of publication October 01, 2015; date of current version October 13, 2015. The work of T. Shu was supported in part by the NSF under Awards CNS-1343156 and CNS-1524931. The work of Y. Chen was supported by the NSF under Awards CNS-1318748 and CNS-0954020 and the Army Research Office under Grant W911NF-13-1-0288. The work of J. Yang was supported in part by the NSF under Award CNS-1318751. A preliminary version of this work was presented at IEEE INFOCOM, Toronto, ON, Canada, 2014.

T. Shu is with the Department of Computer Science and Engineering, Oakland University, Rochester, MI 48309 USA (e-mail: shu@oakland.edu).

Y. Chen is with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: yingying.chen@stevens.edu).

J. Yang is with the Department of Computer Science, Florida State University, Tallahassee, FL 32306 USA (e-mail: jyang5@fsu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2015.2478881

I. INTRODUCTION

THE WIDESPREAD application of location-based services (LBSs) has recently raised serious concerns on the issue of location privacy. In LBS, a mobile user first obtains its location information from a localization infrastructure, and then uses this information to obtain from an LBS server services customized according to its location. 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. For example, by correlating the user's location with the points-of-interest (POIs) the user has visited, one can glimpse into many aspects of the user's personal life [35], including religious belief, health situation, political inclination, hobby affiliation, daily agenda, and so on.

There have been extensive location-privacy studies focused on preventing an LBS server from learning a user's location when the user accesses the server with his location information, e.g., the k-anonymity [16], [13], [27], the mix zones [1], [2], [26], [31], the pseudonym methods [8], [29], [32], and the m-unobservability [4], [5], [18]. While these measures prevent location leakage in accessing the LBS server, they are all 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 is caused by 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' locations as input, and generates the target's location as output, multisided 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) [24], [39], [41]. 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. Note that applying the existing location-privacy LBS mechanisms to localization does not provide a solution to the localization privacy problem. For

example, to protect its location privacy, an anchor may use spatial cloaking, one of the widely used k-anonymity LBS mechanisms, to blur its location when sharing this information with the target. However, based on the blurred/polluted inputs, the target will be unable to calculate its correct location, which should have been the primary goal of the localization process.

While existing research on the localization process is mainly focused on the algorithm's accuracy and energy efficiency, the privacy aspect during the localization process has been largely ignored. There are only a few studies [3], [6], [28], [38], [44] 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 but the target can obtain the anchors' location information, or vice versa. However, none of the existing studies has investigated the privacy leakage issue from the aspects of both the anchors and the target, which become more important in the increasingly pervasive wireless environments. For instance, during the crowdsourcing-based localization [33], [37], 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 may have to be disclosed to the target object. Meanwhile, the target also has a concern on its location privacy, as it 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 address the privacy issues during the localization process 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. Note that here we are focusing on range-based localization methods. For the different category of range-free localization, such as signal-fingerprint-based localization, its privacy issue has been considered in [23].

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. 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 example, our simulations in Figs. 4 and 5 in Section IV-B have shown that for a multi-lateration-based localization [34], by colluding with 4–5 anchors, an adversary will be able to estimate a target's location within an error of 15–30 m. Location uncertainty at this level is usually good enough for an adversary to identify the POI of a target user and thus glimpse into the user's privacy. For instance, in a hospital, with this location resolution, 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 multiparty secure computation problem [11], [15], [42], 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 threefold.

- We propose and formulate the multi-lateral privacy-preserving localization problem as a secure least-squared-error (LSE) estimation for an overdetermined linear system. Different from other applied secure computation that mainly deals with computation between two parties, e.g., inner product between two private vectors [10], [43], ours 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 are originally designed for two parties only. A straightforward extension to multiparty computation will lead to overwhelming computation and communication overhead.
- We exploit the special structure of our problem to develop specialized 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 a 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 to 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) sharing their information so as to help one target mobile device (e.g., laptop, sensor, or tablet), denoted as node 0, to decide its location. The session 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 $\mathbf{x}_i \stackrel{\text{def}}{=} (x_{i1}, \ldots, x_{in})$, 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 allows the anchor to accurately measure d_{0i}, as experimented in [25]. On the other hand, if the separation between the anchor and the target is large (e.g., they are in the same cell), RF ranging [7] will be used. Such a ranging maps a feature of the received signal (e.g., RSS) to communication distance based on certain signal propagation model. Due to uncertainties such as fading and shadowing, greater ranging errors are expected. 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. Our privacy constructions do not depend on the selection of ranging methods.

We use the method of multi-lateration for location calculation [34] due to its simplicity and popularity. The calculation can be performed either by the target itself or by an anchor or a third party that will notify the calculation outcome to the target afterwards. In particular, based on the information of $(\mathbf{x}_i, d_{0i})$ shared by node i'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:

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

where x_{0j}'s are variables. 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. Hence, 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 [34]. In particular, (1) can be rewritten as

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

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

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

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

Rather than solving (2), we focus on the derived linear system, because its linear nature is more amenable to secure computation. The closed-form 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 $\mathbf{x}_0$ (it will be trivial to see that a node will not be able to compute a coarse estimate on an anchor's location). With Level-I privacy, an anchor will be able to compute by itself a coarse estimate of the target's location. Level-II privacy prevents an anchor from making the above 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 location and coarse estimate of the location. We will design a full range of protocols to realize each of the three privacy levels, and we will see that the communication/computation overhead of the protocols increases with the privacy level.

Various use scenarios can be envisioned for the three privacy levels. Specifically, Level-I privacy is suitable for scenarios where the anchors are unlikely to collude (e.g., in a mobile *ad hoc* scenario), and the target does not have a stringent location privacy requirement, so that a coarse estimate by an anchor does not constitute a serious privacy threat to the target. For example, our simulation in Section IV shows that the estimation errors from an independent anchor could reach a few tens of meters, which may be sufficient to prevent the anchor from pinpointing to, e.g., the particular room the target is in. Level-II privacy is suitable for the scenarios where anchors are unlikely to collude and the target has a high requirement on its location privacy. Level-III privacy is the strongest one, and it is suitable for cases where the target requires high location privacy even in the face of anchor collusion.

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, e.g., based on symmetric keys, so that privacy leakage does not come from communication (i.e., no 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 [30], 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 computes $N = pq$ and $\lambda = \mathrm{lcm}(p-1, q-1)$. 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 $\langle N, g \rangle$ and λ, respectively.
- Encryption: Let $m \in \mathbb{Z}_N$ be a plaintext and $r \in \mathbb{Z}_N$ be a random number. The ciphertext is given by $E(m, r) = g^m r^N \bmod N^2$.
- Decryption: Given a ciphertext $c \in \mathbb{Z}_{N^2}$, the corresponding plaintext is obtained by $D(c) = \frac{L(c^{\lambda} \bmod N^2)}{L(g^{\lambda} \bmod N^2)} \bmod N$.

The Paillier cryptosystem has the following useful properties:

- Homomorphic: For any $m_1, m_2, r_1, r_2 \in \mathbb{Z}_N$, we have

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

- Self-blinding:

$$E(m_1, r_1) r_2^N \bmod N^2 = E(m_1, r_1 r_2).$$

The Paillier cryptosystem is semantically secure for sufficiently large N and g. We assume that N and g are 1024 and 160 bits, respectively, for sufficient semantical security [30]. 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. We assume that the key

management for the Paillier cryptosystem is based on existing public key infrastructures (PKI), such as those discussed in [36].

III. Privacy-Preserving Localization Protocols

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. The multi-lateration is based on the multiple ranging results of all the 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 as $d_{0i}^{(k)}$, where $k = 1, \ldots, K$. Following a similar linearization process to that in Section II-B, but this time the cancellation 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

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

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

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 $\Theta_i \stackrel{\text{def}}{=} \mathbf{R}_i^T\mathbf{R}_i$, and $\phi_i \stackrel{\text{def}}{=} \mathbf{R}_i^T\mathbf{s}_i$, where

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

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

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

Protocol Analysis: We now analyze the correctness, privacy, and computation/communication overhead of Protocol 1.

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. These are proved as follows.

The first condition is clear because an anchor i only has its own location and ranging information to construct $\mathbf{R}_i$ and $\mathbf{s}_i$, and therefore computes Θ_i and ϕ_i. For independent nodes, no information is exchanged between anchors. Thus, just based on its own $\mathbf{R}_j$ and $\mathbf{s}_j$, anchor j cannot calculate a different anchor i's location.

The second condition is clear because, based on its own information, an anchor i can only calculate Θ_i and ϕ_i. To calculate node 0's MMSE location, the anchor needs information on Θ_j and ϕ_j, for $\forall j \neq i$, which cannot be obtained by anchor i as there is no information exchange between anchors in Protocol 1. Thus, an anchor cannot calculate $\mathbf{x}_0$.

To prove the third condition, consider the general case that node 0 is trying to calculate anchor i's locations. The only way node 0 could do that is to calculate the location from the information offered by anchor i: the $n \times n$ matrix Θ_i and the $n \times 1$

vector ϕ_i. Node 0 may derive an equation of variable x_{ij}^k's from each element of Θ_i and ϕ_i, getting at most n^2+n independent equations. The independent unknowns in these equations are x_{ij}^k's, for $j=1,\ldots,n$ and $k=1,\ldots,K$, totaling Kn. Therefore, when $K>n+1$, the number of independent variables is greater than the number of independent equations. Thus, node 0 cannot compute anchor i's location.

Combining the above three arguments, Theorem 2 is proved. ■

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 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 another anchor, say anchor i's Θ_i and ϕ_i. However, 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. Thus, such collusion does not increase the number of independent equations that can be used to solve the locations of those noncolluding anchors. Consequently, we may consider the colluded target and anchors as one virtual target. This essentially equals 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 broadcast by node 0, and node 0 only broadcasts this signal once. In this case, 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 $\mathbf{A}_i \stackrel{\text{def}}{=} 2\,[\,x_{m1}-x_{i1} \quad x_{m2}-x_{i2} \quad \ldots \quad x_{mn}-x_{in}\,]$ 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 compute 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

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

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

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

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 $\mathbf{M}_i=[0\cdots-\mathbf{x}_i\ldots 0]^T$ for $i=1,\ldots,m-1$, and $\mathbf{M}_m=[\mathbf{x}_m\ldots\mathbf{x}_m]^T$. Accordingly

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

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

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

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

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

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

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

An observation in (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 $\Omega_i \stackrel{\text{def}}{=} \mathbf{x}_i^T\mathbf{x}_i + \mathbf{P}_i$ and $\psi_i \stackrel{\text{def}}{=} h_i\mathbf{x}_i^T + \mathbf{v}_i$ to the target, and calculates and sends $\alpha_i \stackrel{\text{def}}{=} \mathbf{x}_i^T + \mathbf{w}_i$ and $\beta_i \stackrel{\text{def}}{=} h_i + t_i$ 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 $\Omega_m \stackrel{\text{def}}{=} (m-1)\mathbf{x}_m^T\mathbf{x}_m - \alpha\mathbf{x}_m - \mathbf{x}_m^T\alpha^T + \mathbf{P}_m$ and $\psi_m \stackrel{\text{def}}{=} (m-1)h_m\mathbf{x}_m^T - h_m\alpha - \beta\mathbf{x}_m^T + \mathbf{v}_m$ to the target.
7) Node 0 calculates $\Omega \stackrel{\text{def}}{=} \sum_{i=1}^{m}\Omega_i$ and $\psi \stackrel{\text{def}}{=} \sum_{i=1}^{m}\psi_i$. It then calculates $\mathbf{x}_0^T = \Omega^{-1}\psi$.

Protocol Analysis:

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 $m > n$.

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.

Argument 1 is obvious for anchors $1,\ldots,m-1$because the only information exchanged between any two anchors i and j, where $1 \le i,j \le m-1$, is the random matrices and vectors used to generate $\mathbf{P}_i$'s, $\mathbf{v}_i$'s, $\mathbf{w}_i$'s, and t_i's (corresponding to Steps 1–4, respectively), which are not related to the location of any node. Condition 1 is also true between anchor m and anchor i, $1 \le i \le m-1$. This is because the only way anchor m may conjecture the location of anchor i is to solve the linear equation set constructed from knowing α_i's and β_i's. By treating $\mathbf{x}_i$'s, $\mathbf{w}_i$'s, h_i's, and t_i's, $i = 1,\ldots,m-1$, as variables, the total number of variables is $2(m-1)n+2(m-1)$, whereas the total number of independent linear equations node m may obtain is $(m-1)n+(m-1)+n+1$, where the four terms stand for the number of equations obtained by knowing α_i's, β_i's, the relationship $\sum_{i=1}^{m-1}\mathbf{w}_i = 0$, and $\sum_{i=1}^{m-1} t_i = 0$, respectively. It is clear that when $m > 2$, the number of variables is greater than the number of independent linear equations, and therefore anchor m cannot calculate the location of other anchors.

Argument 2 is true because from (16) and (21) it is clear that calculating the MMSE estimate $\mathbf{x}_0$ requires all Ω_i's and ψ_i's, $i = 1,\ldots,m$. For independent nodes, no single anchor has all the required information. Therefore, no anchor can calculate the location of the target. Moreover, because node i only has its own location and ranging information $\mathbf{x}_i$ and d_{oi} (note that anchor m cannot recover another anchor's location and ranging information from α_i and β_i because of the random mask $\mathbf{w}_i$ and t_i), the best it can do to estimate $\mathbf{x}_0$ is just a random guess within a circle of radius d_{oi} centered at $\mathbf{x}_i$. However, such a random guess is not the type of estimate we have defined in Section II.

Argument 3 can be proved in a similar way to the proof of argument 1. In particular, the only way the target can calculate the location of an anchor is to solve the linear equation set constructed from knowing Ω_i's and ψ_i's. By treating $\mathbf{x}_i$'s, $\mathbf{P}_i$'s, h_i's, and $\mathbf{v}_i$'s, $i = 1,\ldots,m$, as variables, the total number of variables is $nm + n^2m + m + nm$. The total number of independent linear equations the target may obtain is at most $n^2m + nm + n^2 + n$, where the four terms stand for the number of equations obtained by knowing Ω_i's, ψ_i's, the relationship $\sum_{i=1}^{m}\mathbf{P}_i = 0$, and $\sum_{i=1}^{m}\mathbf{v}_i = 0$, respectively. It is easy to see that when $m > n$, the number of variables is greater than the number of equations, and therefore the target cannot calculate the location of any anchor.

Combining the above arguments, Theorem 5 is proved. ■

Theorem 6: When the number of colluding anchors is less than half of $m-1$ and the number of noncolluding 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 noncolluding 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.

For scenario 1, we only need to show that: a) the colluding anchors cannot learn the location of those noncolluding anchors; and b) the colluding anchors cannot learn the location of the target. These can be proved by noting that when the number of colluding anchors is less than half of $m-1$, the colluding nodes cannot compute the $\mathbf{P}_i$'s, $\mathbf{v}_i$'s, $\mathbf{w}_i$'s, and t_i's of those noncolluding anchors. This is because under the way $\mathbf{P}_i$'s, $\mathbf{v}_i$'s, $\mathbf{w}_i$'s, and t_i's are made, as specified in Steps 1–4 of the protocol, the number of linear equations that can be constructed from colluding nodes' $\mathbf{P}_i$'s, $\mathbf{v}_i$'s, $\mathbf{w}_i$'s, and t_i's is smaller than the number of unknowns for noncolluding nodes. Therefore, for Part a), even if node m is colluding (this is the worst case that colluding nodes possess the maximum amount of information), the colluding nodes still have difficulty in computing the location of those noncolluding nodes. In particular, except node m, suppose the number of noncolluding anchors is L. By treating $\mathbf{x}_i$'s, $\mathbf{w}_i$'s, h_i's, and t_i's of those noncolluding anchors as the variables, the total number of variables is $2nL+2L$. The number of independent linear equations the colluding nodes may obtain is at most $Ln+L+n+1$, where the four terms stand for the number of equations obtained by knowing α_i's and β_i's of the noncolluding anchors, and the relationships $\sum_{i=1}^{m-1}\mathbf{w}_i=0$ and $\sum_{i=1}^{m-1}t_i=0$, respectively. It is easy to see that when $L\geq 2$, the number of variables is greater than the number of independent equations, so the colluding nodes cannot compute the location of noncolluding nodes. The proof of Part b) is straightforward: Colluding nodes do not have the full knowledge of all Ω_i's and ψ_i's (for $i=1,\ldots,m$), therefore they cannot compute $\mathbf{x}_0$.

For scenario 2, we only need to show that the colluding nodes cannot compute the location of noncolluding anchors. To do that, we only consider one particular case—node m and the target are among the colluded nodes—the case whereby the colluded nodes possess the maximum amount of information. We show that even under this extreme case, the colluding nodes still have difficulty in computing noncolluding anchors' location, so they will not be able to do it under other (less informative) cases. Specifically, suppose that, except node m, the number of noncolluding anchors is L. By treating the $\mathbf{x}_i$'s, $\mathbf{P}_i$'s, $\mathbf{v}_i$'s, $\mathbf{w}_i$'s, h_i's, and t_i's of those noncolluding anchors as the variables, the total number of variables is $n^2L+3nL+2L$. The total number of independent linear equations obtained by the colluded nodes is at most $n^2L+nL+nL+L+n^2+n+n+1$, where the first four terms stand for the equations obtained by the Ω_i's, ψ_i's, α_i's, and β_i's of the noncolluding nodes, respectively. The last four terms stand for equations obtained from the relationship $\sum_{i=1}^{m}\mathbf{P}_i=0$, $\sum_{i=1}^{m}\mathbf{v}_i=0$, $\sum_{i=1}^{m-1}\mathbf{w}_i=0$, and $\sum_{i=1}^{m-1}t_i=0$, respectively. It is easy to show that when $L>n+1$, the number of variables is guaranteed to be greater than the number of independent linear equations, and therefore the colluded nodes cannot compute the location of noncolluding nodes. Combining the arguments for scenarios 1 and 2, Theorem 6 follows. ■

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. Hence, 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. Hence, 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 pieces of 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. Hence, 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 $\mathbf{h}'_i \stackrel{\text{def}}{=} \begin{bmatrix} 0 \\ \vdots \\ -\sum_{j=1}^{n}x_{ij}^2 \\ \vdots \\ 0 \end{bmatrix}$, for $i=1,\ldots,m-1$,

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

Modifying the definition of h_i in Protocol 2 to $h_i \stackrel{\text{def}}{=} \sum_{j=1}^{n} x_{ij}^2$, for $i = 1, \ldots, m$, it is clear that the first term on the right-hand side (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 $d_i \stackrel{\text{def}}{=} d_{0i}^2 - d_{0m}^2$, $\mathbf{M}_i^T\mathbf{d}$ 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 $d_\Sigma \stackrel{\text{def}}{=} \sum_{j=1}^{m-1} d_i$. Thus, 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 $\psi' \stackrel{\text{def}}{=} \sum_{i=1}^{m} \mathbf{M}_i^T\mathbf{d} = \sum_{i=1}^{m}(\mathbf{M}_i^T\mathbf{d} + \mathbf{Z}_i)$

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 $\mathbf{x}_0^T = \Omega^{-1}(\psi + \psi')$.

Protocol Analysis: We now analyze the correctness, privacy, and computation/communication overhead of Protocol 3.

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 noncolluding 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 if it colludes with other anchors or not; 2) a node cannot compute any other node's location, no matter if it colludes with other nodes or not.

For argument 1, the proof for the independent-node case is straightforward because what an anchor knows is no more than its own location information. To prove the colluding-node case, we point out that the Paillier homomorphic encryption in Steps 2 and 3 of Algorithm 1 prevents an anchor from learning its ranging information. Therefore, colluding anchors only know their own locations. Without knowing the ranging information, colluding nodes cannot compute a rough estimate about $\mathbf{x}_0$.

For argument 2, the case for the independent-node can be proved by noting that Algorithm 1 does not leak any information about an anchor's location, x_{ij}'s, to the target, because of the random mask Z_{ij}'s in Steps 2 and 3 of the algorithm. Hence, the execution of Algorithm 1 in Step 2 of Protocol 3 does not allow the target to calculate any anchor's location. Meanwhile, an anchor will not be able to calculate the location of the target and other anchors because now it has less information than it does in Protocol 2 (ranging information is now owned by the target).

The colluding-node case for argument 2 can be further divided into two subcases: 1) all colluding nodes are anchors; and 2) the target is one of the colluding nodes. For subcase 1, the colluding nodes should not be able to calculate the location of the target and other anchors because now these colluding nodes own less information than they do in Protocol 2 (ranging information is now owned by the target). For subcase 2, we need to show that even under the help of the target, the colluding nodes still cannot calculate the location of other noncolluding anchors. This can be shown by noting that in this subcase the relevant knowledge owned by colluding nodes will be no more than the distance between a noncolluding node and the target. However, just knowing this distance is not enough to calculate the location of the noncolluding node. Except for a random guess, a reasonable estimate on the location of the noncolluding node, as defined in Section II, is also not possible.

Combining the above arguments, Level-III privacy is achieved by Protocol 3. ■

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.

TABLE I
PROTOCOL OVERHEAD

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	$2m\varepsilon_1 + (n+1)m\chi_2 + (2n+1)m\varepsilon_2$	$2048mn + 24[n^2m^2 + (3n+1)m^2]$
HE	Level-II	$2m^3n^2\chi_2 + m^2(m+1)n^2\varepsilon_2 + 2m^3n^2\varepsilon_1$	$2048(m^3n + m^2n^2)$
OT	Level-II	$\mu m^3(n^2 + n)\chi_1$	$\mu m^3 n \times 24$

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. Computation and Communication Overhead

In this section, we compare the computation and communication overhead of the proposed protocols to 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 multiparty secure LSE algorithm that can be applied to compute the MMSE estimate of $\mathbf{x}_0$ [i.e., (6)] in a privacy-preserving fashion. In particular, we consider two well-known algorithms: one based on oblivious transfer (OT) [11] and the other on homomorphic encryption (HE) [17]. 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 [17]. 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 [11], $\mu = 256$ 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 [43], 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 Mb/s.

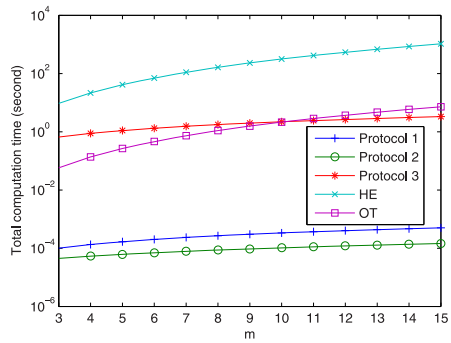


Fig. 1. Computation cost.

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 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 are similar.

1) Numerical Results: We plot the computation cost as a function of the number of anchors in Fig. 1. It can be observed that for Protocols 1–3, 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 1/10–1/100 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 Fig. 2. It can be observed that, for Protocols 1–3, 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

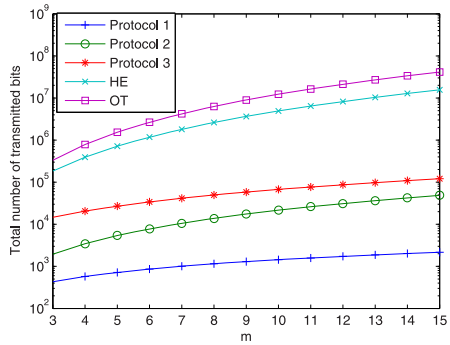


Fig. 2. Communication cost.

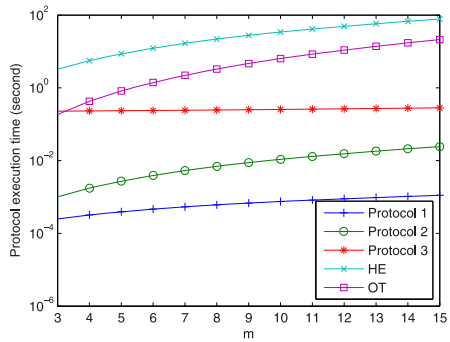


Fig. 3. Protocol execution time.

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 and 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 Fig. 3. Once again, it can be observed that the execution time of the three 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 1–3 ranges from a few milliseconds to hundreds of milliseconds. This indicates that the proposed protocols are very practical. Moreover, it can be observed from Fig. 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.

B. Estimation of Target Location by Collusion

To illustrate how well an anchor can estimate the location of the target by colluding with others, in this section we study the localization accuracy as a function of the number of colluding anchors using computer simulations. We consider a 500 m $\times$ 500 m square area. We assume that the target is located at the center of the square, and the anchors are uniformly randomly distributed in the area. Colluding anchors pool their

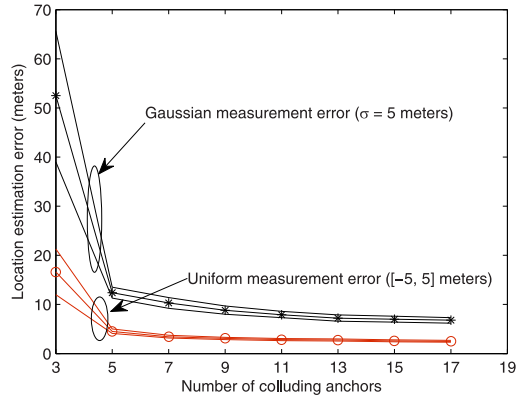


Fig. 4. Localization error (low-ranging error case).

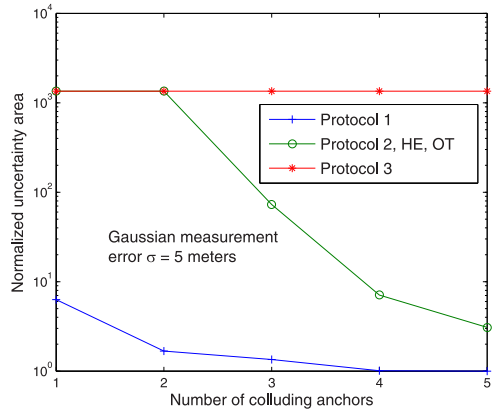


Fig. 5. Localization error (high-ranging error case).

location and ranging information together and use multi-lateration mechanism to estimate the location of the target. A ranging outcome d_{0i} is simulated as the actual distance (d_{0i}^o) plus a ranging error, $d_{0i} = d_{0i}^o + \epsilon$. We consider two distributions for the ranging error ϵ: 1) a Gaussian distribution with 0 mean and standard deviation σ, and 2) a uniform distribution over the range $[-\sigma, \sigma]$. These two distributions may represent unbounded and bounded ranging errors, respectively. For each data point, we performed 500 independent runs and report the average and 95% confidence interval of the results.

The localization errors of the target under low ($\sigma = 5$ m) and high ($\sigma = 15$ m) ranging errors are plotted in Figs. 4 and 5, respectively. Similar trends can be observed in both figures. In particular, under low-ranging errors, for the Gaussian error model, by colluding with four to five other anchors, one can estimate the location of the target with an accuracy of 20 m. For the uniform error model, an anchor can even achieve a 10-m estimation error by colluding with another two anchors. The smaller estimation error in this case is due to the bounded ranging error under uniform distribution. Under high-ranging errors (Fig. 5), the estimation under the same collusion size becomes less accurate. However, in general, an anchor can achieve a 30-m estimation error by colluding with six to seven other anchors. As pointed out in Section I, location resolution at such a level has been sufficient to leak privacy of the target. These observations indicate that it may be unnecessary for an anchor to collude with many others in order to glimpse into the target's privacy—a

small-size collusion, which is relatively easy to form, could be enough. This highlights the necessity of the Level-III privacy achieved by Protocol 3.

C. Location Privacy

As pointed out in Section III, an anchor in the proposed protocols may be able to obtain a coarse estimation of the target's location. In this simulation, we evaluate the target's location privacy resulted from such estimation under the proposed protocols. In particular, we measure the target's location privacy by the uncertainty area in the anchors' estimation, which is a circle centered at the estimated target location and has a radius of the 95th percentile of the localization error (i.e., the true location of the target should be within this uncertainty area with a 95% probability). The larger the uncertainty area, the stronger the location privacy of the target will be. Note that such an uncertainty-area-based privacy measurement is compatible with the commonly used k-anonymity metric, in the sense that it can be converted to the k-anonymity measure by multiplying with the density of nodes in the area.

Our simulations are based on the same setup as in Section IV-B. Moreover, we assume that 20 anchors are used for target localization by Protocol 2, Protocol 3, OT, and HE, respectively. To make a fair comparison, we assume that five mobile anchors are used by Protocol 1, in which each anchor performs ranging at four different places, so that the location estimation by the protocol is also based on 20 pairs of anchor locations and ranging outcomes. In each simulated case, we assume that an arbitrary anchor will try to obtain a coarser estimation of the target location by colluding with a subset of randomly selected anchors. We vary the number of colluding anchors and measure the corresponding uncertainty area in the anchors' estimation based on 500 independent runs. The uncertainty area in the anchors' estimation is normalized w.r.t. the (smaller) uncertainty area in the (more accurate) protocol's estimation.

The normalized uncertainty areas under low ($\sigma = 5$ m) and high ($\sigma = 15$ m) Gaussian ranging errors are plotted in Figs. 6 and 7, respectively. The results under uniform ranging errors have similar trends, and thus are omitted here for concise presentation. Similar trends can be observed in both figures. In particular, it is clear that for Protocols 1–3, their location privacy increases with their level of privacy. This is as expected from the design of these protocols. Specifically, Protocol 1 has the lowest privacy strength because each colluding anchor in Protocol 1 can contribute multiple (4 in the simulation) samples of anchor location and ranging outcome, which greatly improve the estimation accuracy under the same number of colluding anchors. On the contrary, Protocol 3 has the best privacy strength and is immune to anchors' collusion because the collusion of the anchors under this protocol will not leak any ranging information, and thus no estimation on the target location can be made by the colluding anchors. The privacy strength of Protocol 2 is in the middle because each colluding anchor can contribute only one sample of anchor location, and ranging outcome to the estimation, smaller than that of Protocol 1 but greater than protocol 3. The OT and HE methods have the same privacy strength as Protocol 2 because they all

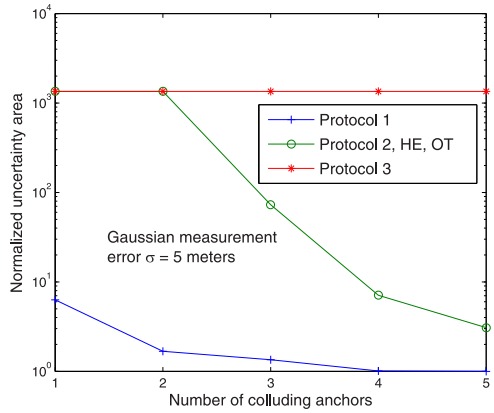


Fig. 6. Target location privacy (low-ranging error case).

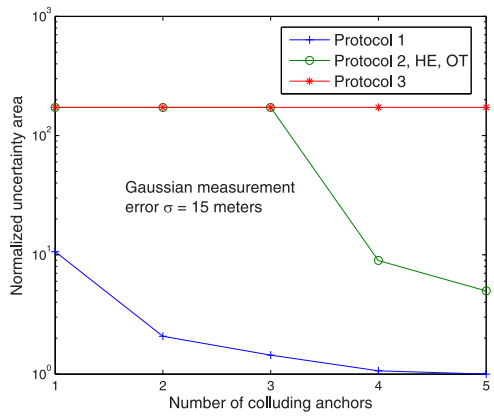


Fig. 7. Target location privacy (high-ranging error case).

achieve Level-II privacy. Moreover, Figs. 6 and 7 also verify that Protocols 1–3 do achieve their intended levels of privacy. Specifically, it shows when an anchor in Protocol 1 tries to estimate the target location by itself, its estimation is significantly coarser (roughly 10 times coarser in the simulation) than that by the protocol, which is desired by the Level-I privacy. Meanwhile, just as required by the Level-II privacy, Protocol 2 can achieve the maximum privacy strength when there is no node collusion (10^3 and 2×10^2 normalized uncertainty areas in Figs. 6 and 7, respectively.). Finally, Protocol 3 always retains the maximum privacy strength irrespective of anchor collusion, conforming with the requirement of the Level-III privacy.

V. Related Work

Despite the large body of work on privacy-preserving access to LBS, only limited results exist on privacy-preserving localization. In particular, localization methods can be broadly classified into two categories: 1) range-based, and 2) range-free. For range-free localization, e.g., those based on signal fingerprints, the localization privacy problem has been addressed in [23] using Paillier homomorphic encryption. Our work belongs to range-based localization. In this category, 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 [20], or by changing the transmission from omnidirectional to a shaped beam using antenna arrays [40]. 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, [38] proposed methods of antenna pattern synthesis to create forged location. Reference [28] extends the effort to multiple mobile nodes by leveraging cooperation among nodes in close vicinity and utilizing synchronized transmissions to obfuscate localization of adversary. On the other hand, unilateral localization privacy is also achieved by the target intentionally injecting a measurement error, which is secretly 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. Reference [44] proposed to induce such a measurement error by manipulating the signal's propagation time. Reference [3] achieves the same goal for an RFID system by controlling RFID tag's response time to the inquiry of the RFID reader.

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 LSE estimation for an overdetermined linear system. Although a secure LSE problem can in general be solved using the classical secure multiparty computation (SMC) techniques [15], e.g., the secure computation circuit method [15], [42], the oblivious transfer method [11], [19], and the method fully based on homomorphic encryption [17], it suffers from high computation/communication cost, which usually renders these methods impractical for real-world problems. To lower the cost, in practice these methods are typically used for two-party computations only, e.g., the secure calculation of set intersection [9], [22], private inner product [14], and the privacy-preserving matching [10], [43]. 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-constrained environment in mobile computing. This renders the existing general SMC techniques unsuitable to our problem. On the other hand, efficient solutions have been proposed for the secure LSE problem based on the centralized commodity-server framework [12], [21], if a trusted central server exists in the computation. Note that such solutions are not applicable to our problem because ours has a distributed setup and no trusted central server can be assumed.

VI. Conclusion

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 the proposed solutions into range-free models, such as those based on signal fingerprints.

References

[1] A. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Comput.*, vol. 2, no. 1, pp. 46–55, Jan.–Mar. 2003.

[2] A. Beresford and F. Stajano, "Mix zones: User privacy in location-aware services," in *Proc. 2nd IEEE Annu. Conf. Pervasive Comput. Commun. Workshops*, 2004, pp. 127–131.

[3] M. Burmester, "Localization privacy," *Cryptography Security*, vol. 6805, Lecture Notes in Computer Science, pp. 425–441, 2012.

[4] Z. Chen, "Energy-efficient information collection and dissemination in wireless sensor networks," Ph.D. dissertation, University of Michigan, Ann Arbor, MI, USA, 2009.

[5] Z. Chen, X. Hu, X. Ju, and K. Shin, "LISA: Location information scrambler for privacy protection on smartphones," in *Proc. IEEE CNS*, 2013, pp. 296–304.

[6] X. Cheng, H. Shu, Q. Liang, and D. H.-C. Du, "Silent positioning in underwater acoustic sensor networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 3, pp. 1756–1766, May 2008.

[7] 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.

[8] Y. K. Choong, "Anonymizing geographic ad hoc routing for preserving location privacy," in *Proc. IEEE ICDCS*, 2005, pp. 646–651.

[9] E. Cristofaro and G. Tsudik, "Practical private set intersection protocols with linear complexity," in *Proc. FC*, Jan. 2010, vol. 6052, pp. 143–159.

[10] W. Dong, V. Dave, L. Qiu, and Y. Zhang, "Secure friend discovery in mobile social networks," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1647–1655.

[11] W. Du and M. J. Atallah, "Privacy-preserving cooperative scientific computations," in *Proc. 14th IEEE Comput. Security Found. Workshop*, Jun. 2001, pp. 273–282.

[12] W. Du and Z. Zhan, "A practical approach to solve secure multi-party computation problems," in *Proc. New Security Paradigms Workshop*, Sep. 2002, pp. 127–135.

[13] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Trans. Mobile Comput.*, vol. 7, no. 1, pp. 1–18, Jan. 2008.

[14] B. Goethals, S. Laur, H. Lipmaa, and T. Mielikainen, "On private scalar product computation for privacy-preserving data mining," in *Proc. 7th Int. Conf. Inf. Security Cryptol.*, 2005, pp. 104–120.

[15] O. Goldreich, "Secure multi-party computation," Working Draft, 2002 [Online]. Available: http://www.wisdom.weizmann.ac.il/home/oded/public_html/foc.html

[16] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. ACM MobiSys*, 2003, pp. 31–42.

[17] R. Hall, A. Rinaldo, and L. Wasserman, "Secure multiparty linear regression based on homomorphic encryption," *J. Official Statists*, vol. 27, no. 4, pp. 669–691, 2011.

[18] P. Hu, K. Xing, X. Cheng, H. Wei, and H. Zhu, "Information leaks out: Attacks and countermeasures on compressive data gathering in wireless sensor networks," in *Proc. IEEE INFOCOM*, May 2014, pp. 1258–1266.

[19] Y. Ishai, M. Prabhakaran, and A. Sahai, "Founding cryptography on oblivious transfer—efficiently," in *Proc. CRYPTO*, 2008, pp. 572–591.

[20] T. Jiang, H. J. Wang, and Y. C. Hu, "Preserving location privacy in wireless LANs," in *Proc. ACM MobiSys*, 2007, pp. 246–257.
[21] J. Kang and D. Hong, "A practical privacy-preserving cooperative computation protocol without oblivious transfer for linear systems of equations," *Int. J. Inf. Process. Syst.*, vol. 3, no. 1, pp. 21–25, 2007.
[22] L. Kissner and D. Song, "Privacy-preserving set operations," in *Proc. CRYPTO*, Aug. 2005, pp. 241–257.
[23] 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.
[24] X. Li, Y. Chen, J. Yang, and X. Zheng, "Achieving robust wireless localization resilient to signal strength attacks," *Wireless Netw.*, vol. 18, no. 1, pp. 45–58, 2012.
[25] H. Liu *et al.*, "Push the limit of WiFi based localization for smartphones," in *Proc. ACM MobiCom*, Aug. 2012, pp. 305–316.
[26] X. Liu *et al.*, "Traffic-aware multiple mix zone placement for protecting location privacy," in *Proc. IEEE INFOCOM*, 2012, pp. 972–980.
[27] J. Meyerowitz and R. R. Choudhury, "Hiding stars with fireworks: Location privacy through camouflage," in *Proc. ACM MobiCom*, 2009, pp. 345–356.
[28] S. Oh, T. Vu, M. Gruteser, and S. Banerjee, "Phantom: Physical layer cooperation for location privacy protection," in *Proc. IEEE INFOCOM*, 2012, pp. 3061–3065.
[29] Y. Ouyang *et al.*, "Providing anonymity in wireless sensor networks," in *Proc. IEEE Int. Conf. Pervasive Services*, Jul. 2007, pp. 145–148.
[30] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. EUROCRYPT*, May 1999, pp. 223–238.
[31] B. Palanisamy and L. Liu, "MobiMix: Protecting location privacy with mix-zones over road networks," in *Proc. IEEE ICDE*, 2011, pp. 494–505.
[32] S. Rahman, M. Mambo, A. Inomata, and E. Okamoto, "An anonymous on-demand position-based routing in mobile ad hoc networks," in *Proc. IEEE SAINT*, 2006.
[33] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. ACM MobiCom*, 2012, pp. 293–304.
[34] 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.
[35] K. Shin, X. Ju, Z. Chen, and X. Hu, "Privacy protection for users of location-based services," *IEEE Commun. Mag.*, vol. 19, no. 1, pp. 30–39, Feb. 2012.
[36] D. R. Stinson, *Cryptography, Theory and Practice*. Boca Raton, FL, USA: CRC Press, 2006.
[37] A. Uchiyama *et al.*, "UPL: Opportunistic localization in urban districts," *IEEE Transt. Mobile Comput.*, vol. 12, no. 5, pp. 1009–1022, May 2013.
[38] T. Wang and Y. Yang, "Location privacy protection from RSS localization system using antenna pattern synthesis," in *Proc. IEEE INFOCOM*, 2011, pp. 2408–2416.
[39] T. Wang and Y. Yang, "Analysis on perfect location spoofing attacks using beamforming," in *Proc. IEEE INFOCOM*, 2013, pp. 2778–2786.
[40] 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 *Proc. CNSR*, 2007, pp. 157–165.
[41] J. Yang and Y. Chen, "Towards attack resistant localization under infrastructure attacks," *Security Commun. Netw.*, vol. 5, no. 4, pp. 384–403, 2012.
[42] A. Yao, "Protocols for secure computations," in *Proc. 23rd Annu. IEEE Symp. Found. Comput. Sci.*, 1982, pp. 160–164.
[43] R. Zhang, J. Zhang, Y. Zhang, J. Sun, and G. Yan, "Privacy-preserving profile matching for proximity-based mobile social networking," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 656–668, Sep. 2013.
[44] 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, 2004.

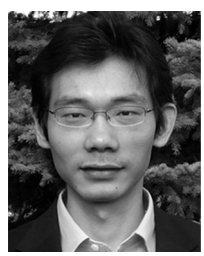
Tao Shu received the B.S. and M.S. degrees in electronic engineering from the South China University of Technology, Guangzhou, China, in 1996 and 1999, respectively, the Ph.D. degree in communication and information systems from Tsinghua University, Beijing, China, in 2003, and the Ph.D. degree in electrical and computer engineering from The University of Arizona, Tucson, AZ, USA, in 2010.

He worked as a Senior Engineer with Qualcomm, Inc., San Jose, CA, USA, from 2010 to 2011. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Oakland University, Rochester, MI, USA. His research aims at addressing the security, privacy, and performance issues in wireless networking systems, with strong emphasis on system architecture, protocol design, and performance modeling and optimization.

Yingying Chen received the Ph.D. degree in computer science from Rutgers University, New Brunswick, NJ, USA, in 2007.

She is a Professor with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA. Prior to joining Stevens, she was with Alcatel-Lucent, Holmdel, NJ, USA. Her research interests include cyber security and privacy, mobile and pervasive computing, and mobile healthcare. She has published over 80 journal and refereed conference papers in these areas.

Prof. Chen is on the editorial boards of the IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and *IEEE Network*. She is a recipient of the NSF CAREER Award and Google Faculty Research Award. She also received an NJ Inventors Hall of Fame Innovator Award. She is the recipient of Best Paper awards from IEEE CNS 2014 and ACM MobiCom 2011. She also received the IEEE Outstanding Contribution Award from the IEEE New Jersey Coast Section each year during 2005–2009. Her research has been reported in numerous media outlets including *MIT Technology Review*, Fox News Channel, *The Wall Street Journal*, and National Public Radio.



Jie Yang (S'08–M'12) received the Ph.D. degree in computer engineering from Stevens Institute of Technology, Hoboken, NJ, USA, in 2011.

He is currently an Assistant Professor with the Department of Computer Science, Florida State University, Tallahassee, FL, USA. His research interests include cyber security and mobile computing, with an emphasis on wireless security, smartphone security and applications, location systems, mobile healthcare, and vehicular applications. His research is supported by the US National Science Foundation (NSF) and Army Research Office (ARO).

Dr. Yang received the Best Paper Award from the IEEE Conference on Communications and Network Security (CNS) 2014 and the Best Paper Award from ACM MobiCom 2011. His research has received wide press coverage including *MIT Technology Review*, *The Wall Street Journal*, NPR, CNET News, and Yahoo! News.