Proximity-Based Positioning Scheme with Multi-layer Privacy

Guillermo Hernandez, Gerald LaMountain, Pau Closas
Northeastern University
Dept. of Electrical & Computer Eng.
Boston, MA (USA)
{hernandez.g, lamountain.g, closas}@northeastern.edu

2023 IEEE/ION Position, Location and Navigation Symposium (PLANS) | 978-1-6654-1772-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/PLANS53410.2023.10140021

***Abstract*—This article investigates the use of proximity-based positioning as a simple, yet effective, positioning system in crowded areas. Particularly, proximity-based positioning is a range-free solution where a receiver computes its position by knowing where nearby receivers are. The main requirement, therefore, is that collaborative receivers need to share their position, which poses a privacy concern and potential limitation in the widespread use of such approach. This article proposes a scheme whereby encrypted positions are shared among the collaborative agents in order to implement a proximity-based solution that does not reveal the position of neighboring nodes. With the use of the homomorphic encryption, we establish a framework that will perform the required operations to obtain position estimates, while the information of any participating user or device will remain private. The concept is investigated and then experimental results provided to support the proposed methodology, showing equivalent performance to the case where no privacy-security guarantees are provided.**

***Index Terms*—Proximity-based service positioning, range-free positioning, collaborative positioning, homomorphic encryption, privacy-preserving.**

I. Introduction

Both outdoor and indoor location and positioning research and technology has increased over the past couple decades. For outdoor tracking, research and technology depend on the satellite technologies, like Global Navigation Satellite System (GNSS) such as, Global Positioning System (GPS) [16]. With an indoor environment, these resources become limited due to physical infrastructures or other obstacles which may lead to signal obstructions or distortion. Addressing the concern of limited GNSS resources for indoor environments has driven researchers to explore methods in order to improve the accuracy of estimating location and position values [2]. Studies presented, and not limited to, in [4], [5], [14], have evaluated different approaches for indoor estimated location and positioning, one of which is the proximity-based service (PBS) method.

It is common to use resources like WiFi for PBS, to be used as alternatives within an indoor setting, as in the case of fingerprint-based indoor position algorithms, proposed in [11]. Other resources like Bluetooth Low Energy (BLE) devices with PBS capabilities can also provide a desired outcome, [7], [27]. Yet, there are other approaches that combine both resources to be able to obtain the common goal, as seen in [24]. Additionally, it is also common to encounter methods that use collaborating users in order to increase the accuracy of these estimated values [23]. An emergence of applications has equally surfaced with these new proposed solutions. As of recently, the urgency of the social-distance measure due to the global pandemic caused by COVID-19 has led to the usage of the PBS, and additionally the increased technology infuse to smart environments [3], [8], [20]

The proximity-based service process requires the necessary step to identify target devices that will partake in a network of devices. With the resources, such as BLE devices or WiFi, the service is able to identify such target devices who may fall within the category to engage in such network in order to optimize the network localization accuracy. In [26], Yin el at. proposed a general framework to establish a threshold for the received-signal-strength (RSS) in order to create such optimal network. As an alternative, other methods include integrating a weighted K-Nearest Neighbor algorithm [19].

Another part of the proximity-based service is the ability to produce accurate estimating position and location values. Just like the initial portion of the service, this area of research has grown and gain much attention. In [21], Subedi el al. proposed using a weighted centroid with affinity propagation clustering to obtain higher accurate estimated values. Bayesian approaches using the Kalman filter, the particle filter, and the non-parametric information filter have also been implemented to remove any noise in the estimated values [15], [25].

These approaches may provide an estimate for position, velocity, and time (PVT) values, which are desired for indoor positioning environments, but these methods lack privacy. The privacy concern is introduced when the sensitive data from collaborative users is used during the computation process when determining the estimated values. An approach that takes into consideration this privacy concern is the KNN classification scheme. It adds a privacy layer by adding noise to the sensitive data used to calculate the PVT estimate values. Based on Chebyshev's inequality, however, this approach would require more resources to validate the estimated values. Thus, more users are required to reduce the variance of the estimated values. Accurate estimation results under these conditions may be infeasible if not enough users are within the service's range of operation.

This work has been partially supported by the National Science Foundation under Award ECCS-1845833.

Another approach to preserve privacy, is the use of encryption. A Fully Homomorphic Encryption has gain much research attention and it was explored in [12] to preserve the privacy between users to obtain PVT estimated values within a cooperative positioning scheme.

In this contribution we propose a two-stage solution which preserves the privacy for a network of users in the proximity-based service. Since, the privacy concern exists in the computational process of estimating the position of a user, it will be referred as the proximity-based positioning scheme. This approach considers all users to be mutually adversarial, or untrustworthy, and it eliminates the need for a trustworthy party that has full access to any sensitive data provided by the network of users. Additionally, it takes into account that the networks users are capable to communicate with one another without any issue.

This contribution is organized as follows. Section II discusses some work related to the paper. Section III, introduces and analyzes the standard proximity-based positioning scheme, where the main assumptions and estimators are discussed. Then, in Section IV, the Fully Homomorphic Encryption methodology is presented and it provides a description of how it is used to encrypt sensitive data, while allowing the scheme to perform the required operations in the encrypted domain. Section V, describes and analyzes the functionality of the proposed encrypted proximity-based positioning scheme. Section VI, contains the evaluation of the estimators analyzed in Section III as used in the proposed privacy-preserving proximity-based positioning scheme. Finally, Section VII, concludes with final remarks and proposes possible extensions to the encrypted proximity-based positioning scheme concept.

II. Related Works

Several works have explored methods of allowing nearby neighbors to exchange their own position in a private manner. One such approach is the k-anonymity method, which obscures the untrusted server rendering it unable to distinguish the true position of a user and $k-1$ other arbitrary positions. The level of privacy under such approach is proportional to the number of users within the network. As the number of k users increases, the privacy level increases. The concept of proximity-based positioning has also been explored in the encryption community. Novak el at. proposed a protocol for identifying a device that is within an indicated distance [18]. Under this protocol, any two users should be willing to share their position with one another once they are within a certain proximity of each other. Both users encrypt their position in such a way that it can be determined if their relative positions are within an indicated subset boundary. If this is determined to be true, the user Bob would send user Alice an n polynomial which Alice must evaluate and send the results back to Bob. Once Bob obtains a result which met the proximity criteria, Bob would send Alice his position.

Additionally, Narayanan et al. proposed using ElGamal encryption in [17] to securely determine the position of a user. The solution enables the users to adjust the proximity area in which they operate and indicates whether or not they are willing to share their position within this area. Additionally, it depends on a social network or graph to establish relationship with people. It uses this information, amongst other technology resources like location tags during the matching process.

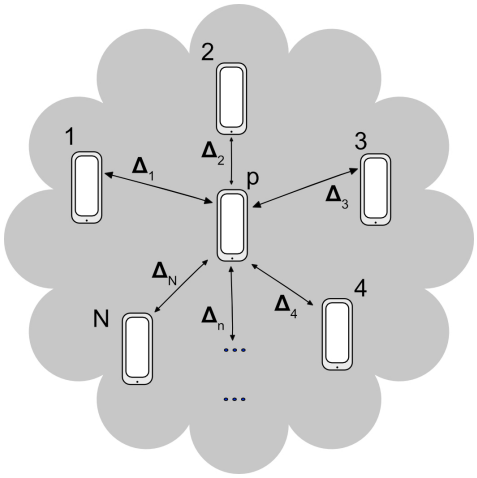

Fig. 1. A Proximity-based positioning scheme consists of multiple neighboring users, sharing their position with the objective of enabling the p-th user to estimate its position.

III. Proximity-based Positioning

This section discusses the mathematical model used in a proximity-based positioning scheme, as well as the main assumptions required for the estimators in this positioning solution. The mismatch between the assumed models and the actual configurations of the network lead to the use of biased estimators, which is discussed here.

Consider a proximity-based positioning scheme which is composed of $N > 2$ neighboring users, who act as mobile devices. This network of users also know their own position coordinates, $\mathbf{m}_n$, and it is known only to themselves, as illustrated in Figure 1. There also exist a p-th user who does not know its position coordinates, $\mathbf{m}_p$. The objective of the network is to provide the p-th user with a position estimate. All N users are positioned relativity near to the p-th user, and the distance between it and an n-th user is $\mathbf{\Delta}_n$. This is modeled as

$$\mathbf{m}_n \simeq \mathbf{m}_p + \mathbf{\Delta}_n \quad 0 < \|\mathbf{\Delta}_n\| < \epsilon \quad n = 1, \ldots, N \tag{1}$$

Assuming that all N users satisfy the condition that the distance between them and the p-th user is less than the boundary condition given by ϵ, then it becomes possible to obtain an estimate of the p-th user's position. However, the n-th user may not guarantee that $\mathbf{m}_n$ is its true position value due to an existing level of uncertainty. The transmitted position (1) is modeled as

$$\mathbf{y}_n = \mathbf{m}_n + \mathbf{w}_n \quad n = 1, \ldots, N \tag{2}$$

where $\mathbf{w}_n$ is a random term that accounts for measurement uncertainty, and this uncertainty is independent of the uncertainty

associated with any other user. $\mathbf{w}_n$ has a Gaussian distribution with zero-mean and finite variance

$$\mathbf{w}_n \sim \mathcal{N}(0, \boldsymbol{\sigma}^2_{\mathbf{w}_n}\mathbf{I}) \tag{3}$$

Again, the network's objective is to estimate the position of the p-th user using the N users observed measurements, also seen as the transmitted position in (2). From all the observed measurements, it becomes possible to construct the likelihood function given the true position of the p-th user, i.e.,

$$p(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N|\mathbf{m}_p)$$

The p-th user can find its position value by maximizing the likelihood function above; this value becomes the estimated position for the p-th user, i.e.,

$$\hat{\mathbf{m}}_p = \arg\max_{\mathbf{m}_p} p(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N|\mathbf{m}_p) \tag{4}$$

The estimated position values' accuracy will highly depend on the amount of assumptions made during the process of maximizing the likelihood function, thus leading to the different possible estimators for the proximity-based positioning scheme. Across all the estimators, it is assume that the uncertainty in the measurements seen in the observed data have the same distribution and they are independent from one another, therefore the observed data values have an iid.

First, consider the case where each n-th user knows the variance of its measurement uncertainty and this uncertainty is not the same for all N users (i.e., $\boldsymbol{\sigma}^2_{\mathbf{w}_n} \neq \boldsymbol{\sigma}^2_{\mathbf{w}_{n'}}$ for $n \neq n'$). Additionally, there exists a distance between the n-th user and the p-th user (i.e., $\boldsymbol{\Delta}_n \neq 0$). This will be considered the optimal estimator and is given by,

$$\hat{\mathbf{m}}_p = \frac{\sum_{n=1}^{N}(\mathbf{y}_n - \boldsymbol{\Delta}_n)(\boldsymbol{\sigma}^{-2}_{\mathbf{w}_n}\mathbf{I})}{\sum_{n=1}^{N}(\boldsymbol{\sigma}^{-2}_{\mathbf{w}_n}\mathbf{I})} \tag{5}$$

It is not always possible that a user may know the variance of its measurement uncertainty, therefore this will change the estimator above. This leads to the next estimator.

The unweighted optimal estimator considers that each measurement uncertainty variance for the observed data is the same for all the data received, meaning that $\boldsymbol{\sigma}^2_{\mathbf{w}_n} = \boldsymbol{\sigma}^2_{\mathbf{w}_{n'}}$ for $n \neq n'$, but it still considers that each n-th user's position is some distance away from the p-th user and is given by,

$$\hat{\mathbf{m}}_p = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}_n - \boldsymbol{\Delta}_n) \tag{6}$$

Similar to the optimal estimator (5), the unweighted optimal estimator (6) assumes that each n-th user knows the p-th user's position, since it knows the distance from the p-th user. This also considers that the $\boldsymbol{\Delta}_n \neq 0$ for all N users.

The last estimator is the unweighted proximity-based estimator. It has the same assumptions as the unweighted optimal estimated in (6) and it extends on the assumption that as the distance between the n-th and p-th users approaches zero and the n-th user's position converges to the p-th user's position, i.e., $\mathbf{m}_n \simeq \mathbf{m}_p$ as $\boldsymbol{\Delta}_n \to 0$. The estimator is given by,

$$\hat{\mathbf{m}}_p = \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n \tag{7}$$

$$E[\hat{\mathbf{m}}_p] = \frac{1}{N}\sum_{n=1}^{N}E[\mathbf{y}_n] \overset{\boldsymbol{\Delta}_n \to 0}{=} \frac{1}{N}\sum_{n=1}^{N}\mathbf{m}_p = \mathbf{m}_p \tag{8}$$

It is noted that the unweighted proximity-based estimator is consider to be an unbaised estimator, as shown in (8).

The unweighted proximity-based estimator will be considered later for the encrypted proximity-based positioning scheme.

IV. Homomorphic Encryption

The purpose of this article is to design a privacy-preserving scheme for proximity-based positioning, which necessitates sharing of position information among the network of users. Here we review the fundamental aspects of fully homomorphic encryption as used in Section V. Research on the topic of Fully Homomorphic Encryption (FHE) has rapidly increase, since the first solution was introduced by Gentry [9], [10], [22]. The significance of FHE is due to its ability to perform unlimited number of addition and multiplication operations on ciphertexts, a capability that was not possible before [1]. Since its first proposed solution, many have used the Gentry blueprint to develop newer FHE schemes based on the computation security of the Learning with Errors (LWE) problem and the ring-LWE problem. FHE schemes come with a high computationally expense and this has led to different attempts to minimize this issue, such as in [6], [10].

A FHE scheme requires three main components: a key generating algorithm $\mathsf{KeyGen}(\cdot)$, an encryption algorithm $\mathsf{Enc}(\cdot)$, and a decryption algorithm $\mathsf{Dec}(\cdot)$. The $\mathsf{SecretKeyGen}$ and $\mathsf{PublicKeyGen}$ algorithms, which fall under the key generating algorithm $\mathsf{KeyGen}(\cdot)$, are responsible for creating a pair of private and public keys, respectively. The public key is used to encrypt a plaintext message into a ciphertext, and the private key is used to decrypt a ciphertext into a plaintext message. If a primary user wishes to communicate with a secondary user, the primary user will generate a set of public and private keys. The public key is distributed to the secondary user. This secondary user may then use the public key to encrypt its message before sending it to the primary user. The primary user then receives the encrypted message (in ciphertext form) and uses its private key for the decryption process. After the decryption process, the primary user will obtain the seondary user's message. The private key always remains with the primary user and it is never distributed to any other user.

Cryptographic homomorphism refers to encryption methods that allow certain operations to be performed on encrypted data. A fully homomorphic encryption (FHE) system is one that supports performing an unlimited number of addition and multiplication operations on encrypted data without corrupting

the value obtained when the result is decrypted. This means that if the encryption and decryption functions are denoted as $\text{Enc}(\cdot)$ and $\text{Dec}(\cdot)$, and the operator is represented by $f(\cdot,\cdot)$, then for two messages m_1 and m_2 we have:

$$\text{Dec}(f(\text{Enc}(m_1), \text{Enc}(m_2))) = f(m_1, m_2) \tag{9}$$

This allows for multi-step computations to be performed on encrypted data and it will produce the same result as if the data were not encrypted. Since FHE also allows for an unlimited number of operations to be performed on encrypted data, it is usable in the deployment of more complex algorithms and applications.

As briefly mentioned, the FHE systems are considered to be secure based on the computational security of the LWE problem, or its variant, the RLWE problem. The security of the system is determined by the security parameter, which are commonly 128-bit, 192-bit, or 256-bit. For a FHE system based on the RLWE problem, such as the FV scheme in [6], messages are converted into plaintexts, which are then encrypted into ciphertexts. The plaintext space is taken from a polynomial modulus quotient ring R_t with polynomial degree n and coefficients with modulus t. Similarly, the ciphertext is a polynomial in R_q with modulus value q. Based on [6], the polynomial degree, n, must be a power of 2. A larger n value leads to a larger polynomial size and it increases the computation resources. Additionally, with a large polynomial degree, there is an increase in the ciphertext and plaintext modulus values, q and t, respectively.

V. Encrypted Proximity-based Positioning Network

This section provides an overview of the proposed privacy-preserving proximity-based positioning scheme. Important characteristics of the proposed protocol, such as unique users that will be critical for the protocol, the exchange of information, and the application of FHE, are all defined.

A. Overview

The proposed privacy-preserving proximity-based positioning scheme uses two layers of encryption to maintain the privacy for all the users within the network. Figure 2 illustrates the establishment of these encryption layers, the necessary inputs and outputs of each layer, and the public and private keys needed to access these layers.

A layer of encryption is created with a set of public and private keys. The case of a two layer of encryption, it will required two sets of public and private keys. A single layer of encryption would not be appropriate for the privacy-preserving proximity-based positioning scheme. This will introduces the possible risk that the user who created the first layer, will have access to the set of public and private keys. This will indicate that this user will have access to all the encrypted data, in this case the positions of all the users, and it will be able to decrypt all their encrypted messages. However, by having two layers, this approach will keep the users who have access to the private keys accountable, and any sensitive data

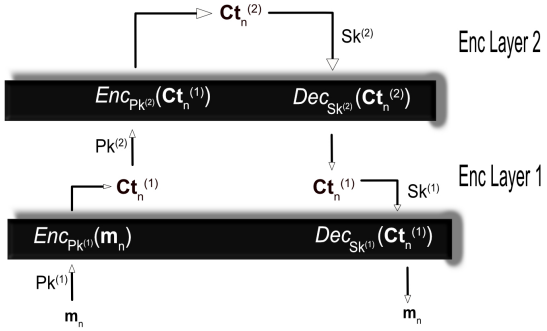

Fig. 2. The message, $\mathbf{m}_n$, is encrypted twice in order to preserve it privacy. $\mathbf{m}_n$ is first encrypted using the first public key, $\mathbf{Pk}^{(1)}$, then its ciphertext within the first layer of encryption, $\mathbf{Ct}_n^{(1)}$ is encrypted using the second public key, $\mathbf{Pk}^{(2)}$. This is the ciphertext within the second layer of encryption, $\mathbf{Ct}^{(2)}$. To decrypt the message, a similar process is performed. The second private key, $\mathbf{Sk}^{(2)}$, decrypts $\mathbf{Ct}^{(2)}$ to obtain a ciphertext within the first layer of encryption, $\mathbf{Ct}_n^{(1)}$ and $\mathbf{Sk}^{(1)}$ used to decrypt $\mathbf{Ct}_n^{(1)}$ to reveal the message $\mathbf{m}_n$.

will remain private throughout the entire process. Therefore, the most appropriate approach is to incorporate two layers of encryption.

Additionally, it is important that two unique users create these layers of encryption, one user for each layer. If a single user creates the two layers of encryption, it will obtain access to the private key for both layers of encryption. This result is the equivalent to having a single layer of encryption. Since the user will have access to the private key for both layers of encryption, this will give the user the ability to decrypt the doubly encrypted message. To address this issue, one user creates a single layer of encryption and a different user creates the second layer of encryption to maintain the integrity of the scheme.

Furthermore, any user within the network of users may perform the encrypted computations. For simplicity, it is determined that the user who creates the second layer of encryption will perform the encrypted computation.

Following the illustration of Fig 2, the input for the first layer of encryption is a plaintext message, $\mathbf{m}_n$; this is the n-th user's position values. Using the public key for the first layer of encryption (i.e., $\mathbf{Pk}^{(1)}$), $\mathbf{m}_n$ is converted into a ciphertext, i.e., $\mathbf{Ct}_n^{(1)}$. Subsequently, $\mathbf{Ct}_n^{(1)}$, becomes the input to the second layer of encryption. Using the public key for the second layer of encryption ($\mathbf{Pk}^{(2)}$), each element that comprises $\mathbf{Ct}_n^{(1)}$ are encrypted, thus producing a ciphertext for each element. The outcome is a total of $(K \times L)$ ciphertexts, where $(K \times L)$ is the dimension of $\mathbf{Ct}_n^{(1)}$, and K and L depend on the encrypted parameters set during the key development process. For simplicity, let $\mathbf{Ct}_n^{(2)}$ represent the set of ciphertext within the second layer of encryption.

Traversing the opposite direction will require the appropriate private key to obtain the message or ciphertext contained in these ciphertexts. Starting with the second layer of encryption, the second layer's private key (i.e., $\mathbf{Sk}^{(2)}$) is required to decrypt the message within the ciphertexts $\mathbf{Ct}_n^{(2)}$. When the

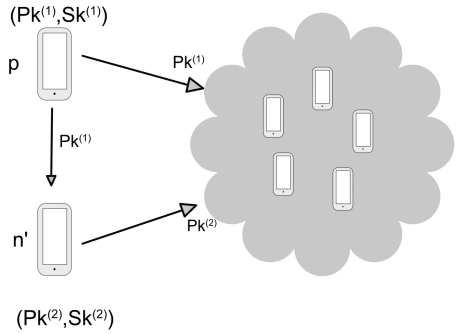

Fig. 3. The p-th user creates the first pair of keys for the encryption, and the n'-th user creates the second pair of keys. Both users distribute their public keys to all N users, except the n'-th user does not need to distribute its public key to the p-th user.

decryption process finishes, the decrypted result is a set of coefficients that comprises the ciphertext $\mathbf{Ct}_n^{(1)}$. Then, the first layer's private key, $\mathbf{Sk}^{(2)}$, is used to decrypt $\mathbf{Ct}_n^{(1)}$. After the decryption process, the decrypted result is the encrypted message. Since the n'-th user created the second layer of encryption, this will be the same user who will perform the encrypted computations.

B. Distribution of Keys

The encrypted proximity-based positioning scheme requires two users to create their own the public and private keys. The first user, the p-th user, is the user whose position is unknown as seen in Fig. 1. The second user, the n'-th user, is part of the set of users that exist near the p-th user. The second user is chosen at random and $n' \in N$.

As seen in Fig. 3, the distribution of the keys is performed in two steps. First, the p-th user creates the first set of keys, ($\mathbf{Pk}^{(1)}$, $\mathbf{Sk}^{(1)}$) and the user distributes $\mathbf{Pk}^{(1)}$ to all the users, including the n'-th user. Simultaneously, the p-th user creates the first layer of encryption. However, the p-th user retains its exclusive access to $\mathbf{Sk}^{(1)}$. With this access, the p-th user is capable of decrypting any ciphertext within the first layer of encryption. This will become important when the p-th user obtains its estimated position values, thus the p-th user must always create the first layer of encryption.

The n'-th user creates the second set of keys, ($\mathbf{Pk}^{(2)}$, $\mathbf{Sk}^{(2)}$) during the second step. The n'-th user also distributes its public key to all N users, with the possible exception of the p-th user. The n'-th user holds its access to the private key $\mathbf{Sk}^{(2)}$. Thus, the n'-th use creates the second layer of encryption and it has the capability to decrypt all the ciphertexts within the second layer of encryption.

C. Encryption Process

After receiving the two public keys, ($\mathbf{Pk}^{(1)}$, $\mathbf{Pk}^{(2)}$), each user within the network of users encrypts its position data. First, they encrypt their data using $\mathbf{Pk}^{(1)}$, to create the first ciphertext $\mathbf{Ct}_n^{(1)}$, that is

$$\mathbf{Ct}_n^{(1)} = \mathrm{Enc}_{\mathbf{Pk}^{(1)}}(\mathbf{m}_n) \qquad n = 1, \ldots, N \tag{10}$$

where $\mathbf{m}_n$ represents the position values for the n-th user. Each user creates this ciphertext locally and they do not transmit this ciphertext to the n'-th user. Instead, each user uses the second public key, $\mathbf{Pk}^{(2)}$ to encrypt $\mathbf{Ct}_n^{(1)}$ in order to create a ciphertext within the second layer of encryption, $\mathbf{Ct}_n^{(2)}$, as follows,

$$\mathbf{Ct}_{(n;k,\ell)}^{(2)} = \mathrm{Enc}_{\mathbf{Pk}^{(2)}}\left(\mathbf{Ct}_n^{(1)}(k,\ell)\right) \tag{11}$$

where, $n = 1, \ldots, N$, $k = 1, \ldots, K$, and $\ell = 1, \ldots, L$.

This is equivalent to encrypting $\mathbf{m}_n$ twice, once using $\mathbf{Pk}^{(1)}$ then using $\mathbf{Pk}^{(2)}$, i.e.,

$$\mathbf{Ct}_n^{(2)} = \mathrm{Enc}_{\mathbf{Pk}^{(2)}}\big(\,(\mathrm{Enc}_{\mathbf{Pk}^{(1)}}(\mathbf{m}_n))\,\big) \qquad n = 1, \ldots, N$$

This second layer of encryption requires more than one ciphertext in order to preserve the privacy of the position data. From (11), a second layer of encryption will require $K \cdot L$ ciphertexts from a single user. This indicates that the (k^{th}, ℓ^{th}) element of $\mathbf{Ct}_n^{(1)}$ needs to be encrypted using $\mathbf{Pk}^{(2)}$. After creating all the required ciphertexts within the second layer of encryption, the users transmit these ciphertexts to the n'-th user to perform the encrypted computation.

The n-user can transmit its doubled-encrypted sensitive data to the n'-th user, knowing that the n'-th user does not have access to $\mathbf{Sk}^{(1)}$, which is required to decrypt the first layer of encryption. In the case that the p-th user intercepts $\mathbf{Ct}_n^{(2)}$, the n-th user's data will still remain private due to the fact that the p-th user does not have access to $\mathbf{Sk}^{(2)}$, which is required to decrypt the second layer of encryption. Therefore, the n-th user's sensitive data will remain private at all time.

D. Encryption Operations

The ciphertexts within the second layer of encryption of all the users are received by the n'-th user, i.e. $\{\mathbf{Ct}_{1;1:K,1:L}^{(2)}, \ldots, \mathbf{Ct}_{N;1:K,1:L}^{(2)}\}$. In total there will be $N \cdot K \cdot L$ ciphertexts received. Recall the n' user contains the private key for the second layer of encryption, $\mathbf{Sk}^{(2)}$. This will be of no use to decrypted the plaintext computation results that occurs within the second layer of encryption, because $\mathbf{Sk}^{(1)}$ is required and only the p-th user has access to it. The n'-th user will perform the required encrypted computation using all the received ciphertexts, thus specifying the second layer of encryption to be the layer of encrypted computation. Using the estimator in (7), the n'-th user will perform the summation of all the users encrypted data. Since each ciphertext $\mathbf{Ct}_n^{(1)}$, maintains the same dimensions, the n'-th user will need to add the elements that shared the same index across all the users' ciphertext from the first layer of encryption $\mathbf{Ct}_n^{(1)}$ as follows,

$$\mathbf{Ct}_{(s,k,\ell)}^{(2)} = \sum_{n=1}^{N}\left(\mathbf{Ct}_{(n,k,\ell)}^{(2)}\right) \quad k = 1, \ldots, K \quad \ell = 1, \ldots, L \tag{12}$$

where $\mathbf{Ct}^{(2)}_{(s,1:k,1:L)}$ is the encrypted computation solution to the second layer of encryption. The result of (12) is consistent of adding the corresponding entries for all the N first level ciphertexts, and then encrypting the result, i.e

$$\sum_{n=1}^{N} \mathbf{Ct}^{(2)}_{(n,k=1,\ell=1)} = \sum_{n=1}^{N} \mathrm{Enc}_{\mathbf{Pk}^{(2)}} \left(\mathbf{Ct}^{(1)}_{n}(k=1,\ell=1) \right) = \mathrm{Enc}_{\mathbf{Pk}^{(2)}} \left(\sum_{n=1}^{N} \mathbf{Ct}^{(1)}_{n}(k=1,\ell=1) \right)$$

The procedure will be the same for all K and L entries. In total, there will be $N \cdot K \cdot L$ encrypted calculations performed. If each ciphertext that forms $\mathbf{Ct}^{(2)}_{s}$ is decrypted, the result will be $K \cdot L$ values, which forms all the elements of a first level ciphertext. Therefore, when (12) performs all the computations in the second layer of encryption, this becomes the equivalent of the summation all the users position values, i.e.,

$$\mathbf{Ct}^{(2)}_{s} = \sum_{n=1}^{N} \mathrm{Enc}_{\mathbf{Pk}^{(2)}} \left(\mathrm{Enc}_{\mathbf{Pk}^{(1)}} \left(\mathbf{m}_n \right) \right) = \mathrm{Enc}_{\mathbf{Pk}^{(2)}} \left(\mathrm{Enc}_{\mathbf{Pk}^{(1)}} \left(\sum_{n=1}^{N} \mathbf{m}_n \right) \right)$$

E. Decryption Process

After performing (12), the results remain encrypted within the second layer of encryption. The n'-th user proceeds to decrypt the encrypted result using its private key, $\mathbf{Sk}^{(2)}$, to produce a ciphertext within the first layer of encryption, as follows,

$$\mathbf{Ct}^{(1)}_{s}(k,\ell) = \mathrm{Dec}_{\mathbf{Sk}^{(2)}} \left(\mathbf{Ct}^{(2)}_{s,k,\ell} \right) \tag{13}$$

where $k = 1, \ldots, K$, $\ell = 1, \ldots, L$, and $\mathbf{Ct}^{(1)}_{s}$ is the decrypted result of the second layer of encryption. The result produces a single ciphertext acquiring the same number of entries as before and more importantly, the results reflects the desire summation result.

$$\mathbf{Ct}^{(1)}_{s}(k=1,\ell=1) = \mathrm{Dec}_{\mathbf{Sk}^{(2)}} \left(\mathbf{Ct}^{(2)}_{s,k=1,\ell=1} \right) = \mathrm{Dec}_{\mathbf{Sk}^{(2)}} \left(\mathrm{Enc}_{\mathbf{Pk}^{(2)}} \left(\sum_{n=1}^{N} \mathbf{Ct}^{(1)}_{n}(k=1,\ell=1) \right) \right) = \sum_{n=1}^{N} \mathbf{Ct}^{(1)}_{n}(k=1,\ell=1)$$

The approach above can be repeated for all the K and L entries to attest that the decrypted results in (13) is equivalent to summation of the corresponding entries of the non-second-layer-encryption result. Additionally, (13) also corresponds to the n'-th user removing the second layer of encryption, i.e.,

$$\mathbf{Ct}^{(1)}_{s} = \mathrm{Dec}_{\mathbf{Sk}^{(2)}} \left(\mathrm{Enc}_{\mathbf{Pk}^{(2)}} \left(\mathrm{Enc}_{\mathbf{Pk}^{(1)}} \left(\sum_{n=1}^{N} \mathbf{m}_n \right) \right) \right) = \mathrm{Enc}_{\mathbf{Pk}^{(1)}} \left(\sum_{n=1}^{N} \mathbf{m}_n \right)$$

After decrypting the results and obtaining the ciphertext within the first layer of encryption, the n'-th user transmits $\mathbf{Ct}^{(1)}_{s}$ to the p-th user. The p-th user is the user who created the pair of keys for the first layer of encryption, thus indicating that the user has $\mathbf{Sk}^{(1)}$. With this key, the p-th user is able to decrypt $\mathbf{Ct}^{(1)}_{s}$.

The p-th user proceeds to decrypt the received ciphertext using its private key $\mathbf{Sk}^{(1)}$,

$$\bar{\mathbf{m}} = \mathrm{Dec}_{\mathbf{Sk}^{(1)}} \left(\mathbf{Ct}^{(1)}_{s} \right) = \mathrm{Dec}_{\mathbf{Sk}^{(1)}} \left(\mathrm{Enc}_{\mathbf{Pk}^{(1)}} \left(\sum_{n=1}^{N} \mathbf{m}_n \right) \right) = \sum_{n=1}^{N} \mathbf{m}_n \tag{14}$$

The result from (14), is the summation of position for all the users in the network. Now, the p-th user can obtain its estimated position values using (14) and given that number of users in the network are known, ie,

$$\hat{\mathbf{m}}_p = \frac{1}{N} \bar{\mathbf{m}} \tag{15}$$

Throughout the entire process, the sensitive data sent by all N users remained private. Due to the ability to have the data encrypted twice by using different public keys, this prevented the n'-th user to obtain the computational result after it performed the operations. As mentioned before, the n'-th user is not required to compute the encrypted computation seen in Section V-D. Any user within the network of users or even the p-th user may perform these computations, but the encrypted result in the second layer of encryption must always be returned to the n'-th user. This is due to the n'-th user having access to $\mathbf{Sk}^{(2)}$.

The analysis of the encrypted proximity-based positioning scheme is able to produce the estimator seen in (7), where the noise variance is constant for all users and $\mathbf{m}_n \simeq \mathbf{m}_p$ as $\mathbf{\Delta}_n \to 0$ for $n = 1, \ldots, N$. Furthermore, this shows that compared to the estimator in (6), the unweighted proximity-based estimator is a more realistic estimator, since (6) assumes that the n-th user knows its distance from the p-th user. This cannot be the case, since at all time, the p-th user and the n-th user positions are kept private from each other. Additionally, we note the importance of computation complexity. The estimator seen in (5) will require multiple multiplication operations. Within the FHE domain, this will drastically increase the space overhead and the computation complexity. Therefore, in the environment of the encrypted proximity-based positioning scheme, the estimator that requires the least amount of computational operations would reduce the amount of space and time that a user needs to make the calculations, thus making (7) the ideal estimator.

VI. Results

The results are divided into two parts. The first analyzes the proximity-based position estimator, while the second set of experiments aim at analyzing the proposed proximity-based encrypted positioning solution.

A. Proximity-based Estimator

To analyze the three different estimators, seen in (5), (6), and (7), we created a simulator that generator random data (position coordinates) around a stationary position. The data was generated according to

$$\mathbf{y}_n \sim \mathcal{N}(\mathbf{m}_p + \mathbf{\Delta}_n, \boldsymbol{\sigma}_n^2 \mathbf{I}) \tag{16}$$

The mean of the distribution consisted of the "true" or stationary position of the p-th user and the deterministic distance between the p-th user and the n-th user. This deterministic distance is represented by $\mathbf{\Delta}_n$. Furthermore, the variance value for each user was different, thus $\boldsymbol{\sigma}_{n-1}^2 \neq \boldsymbol{\sigma}_n^2$.

The boundary set for the proximity-based positioning scheme was set to $\epsilon = 40$ (m), as seen in (1). Any user outside this boundary was not considered to be part of the network of users.

We evaluated each estimator using different numbers of observations performed by a certain set of users. For example, when the simulator set 5 users to be present, it collected a single sample and determined the RMS error, then a second time the simulator would observed 2 samples for the same 5 users and then determine the RMS error. The maximum number of observations record for a certain set of users was 100 measurement values and there were at most 50 users present. A summary of the RMSE values are given in Fig. 4.

From the results given in Fig. 4, it is clear that the optimal estimator seen in (5) performed the best compared to the other two estimators. This estimator did not make much assumptions from the sample random generator, it had the information of the true distance between the p-th user and the n-th user. This means that the estimator did not consider $\mathbf{\Delta}_n$ to be zero. Furthermore, it also had acknowledge of true values of each random sample.

The next best estimator is the estimator seen in (6), also seen in Fig. 4 as the unweighted optimal estimator. Just like the optimal estimator, the unweighted optimal estimator had knowledge of the distance between the p-th user and the n-th user, meaning $\mathbf{\Delta}_{n-1} \neq \mathbf{\Delta}_n$. With this information, the estimator was able to minimize the impact from users that were further away from the p-user. The difference seen the between the unweighted optimal estimator and the optimal estimator is established through the knowledge of the variance of the measurement uncertainty. Since the unweighted optimal estimator had zero knowledge of the variance, it assumed that every n-th user measurement uncertainty variance is the same.

The unweighted proximity-based estimator had the most restrictive assumptions; every n-th user's variance is the same and they share the same position as the p-th user. Based on the results in Fig 4, these estimators will improve as more users are within the network.

B. Encryption Parameters

Based on the results in section VI-A, we analyzed the unweighted proximity-based estimator to implement with the fully homomorphic encryption, using the Pyfhel library [13],

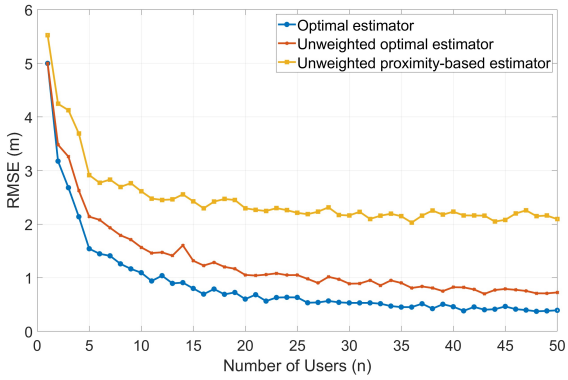

Fig. 4. RMSE performance as a function of the number of collaborative users n for the 'Optimal Estimator' in (7), the 'Unweighted optimal estimator' in (5), and the 'Unweighted proximity-based estimator' in (8).

We tested the different encryption parameters, mainly the polynomial degree value n. We saw this parameter to be crucial for our solution because of its influence in the computational and storage cost.

As we saw in section IV, the polynomial degree is a factor on the modulus value for the ciphertext modulus value and it had the requirement that the polynomial degree needs to be a value that is a power of 2. The higher the polynomial degree, the more the ciphertext modulus domain grower. Similarly, this was the same situation for the plaintext modulus value.

We first tested the case when the polynomial degree value was set to 1024. This value was set to both layers of encryption. It is important to note, because of the limitation this value was not successful with the second layer of encryption. This second layer of encryption required a higher plaintext modulus because of the ciphertext coefficient values were high and the amount of computation seen during the computation process was excessive for the domain.

Second, we set the polynomial degree to 2048. These results are seen in Fig. 5. Based on these results, the number of users that participate within the solution may increase the computational complexity. As seen earlier, an estimator may improve accuracy with more users or target devices engage in the network, but it may hinder performance when addressing the privacy concern issue. Therefore, increasing the accuracy will increase the computational cost. The third polynomial degree tested was 4096. At this value, the ciphertexts grew exponentially, and it required a large amount of memory storage. It can be argue that this may become ideal to improve accuracy, but it also increases the computational cost.

VII. Conclusion

This article investigated the use of proximity-based for range-free positioning of devices. In particular, the contribution of this work is to present a novel scheme that enables such positioning in a privacy-preserving manner such that collaborative agents in the network do not reveal their position in the process. The proposed framework uses fully homomorphic encryption methodology, where certain operations

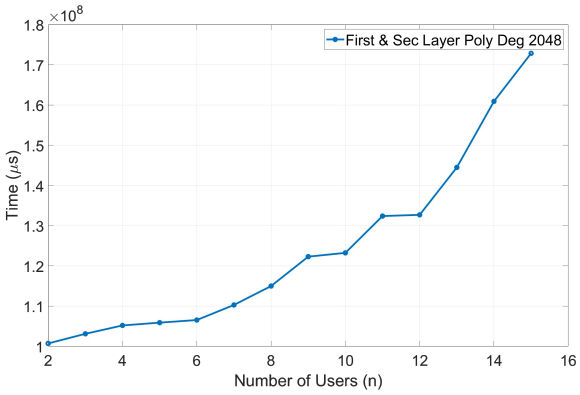

Fig. 5. Encrypted Proximity-based positioning run-time with polynomial degree n set to 2048.

can be perform on encrypted data. We introduced a multi-layer encryption scheme to implement the privacy-preserving proximity-based position scheme system which achieves the objective of preserving position information of agents to be revealed. We identified that two layers would provide the sufficient about of privacy required and it keeps every participate accountable in the duration of time. Based on the results, the proposed scheme will be ideally used with a polynomial degree small enough that will provide the privacy for the users.

REFERENCES

[1] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (CSUR)*, 51(4):1–35, 2018.

[2] M. G. Amin, P. Closas, A. Broumandan, and J. L. Volakis. Vulnerabilities, threats, and authentication in satellite-based navigation systems [scanning the issue]. *Proceedings of the IEEE*, 104(6):1169–1173, 2016.

[3] V. Bianchi, P. Ciampolini, and I. De Munari. Rssi-based indoor localization and identification for zigbee wireless sensor networks in smart homes. *IEEE Transactions on Instrumentation and Measurement*, 68(2):566–575, 2018.

[4] D. Dardari, P. Closas, and P. M. Djurić. Indoor tracking: Theory, methods, and technologies. *IEEE Transactions on Vehicular Technology*, 64(4):1263–1278, 2015.

[5] P. Davidson and R. Piché. A survey of selected indoor positioning methods for smartphones. *IEEE Communications Surveys & Tutorials*, 19(2):1347–1370, 2016.

[6] J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. *Cryptology ePrint Archive*, 2012.

[7] R. Faragher and R. Harle. Location fingerprinting with bluetooth low energy beacons. *IEEE journal on Selected Areas in Communications*, 33(11):2418–2428, 2015.

[8] M. Fazio, A. Buzachis, A. Galletta, A. Celesti, and M. Villari. A proximity-based indoor navigation system tackling the covid-19 social distancing measures. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE, 2020.

[9] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009.

[10] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Annual Cryptology Conference*, pages 75–92. Springer, 2013.

[11] S. He and S.-H. G. Chan. Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys & Tutorials*, 18(1):466–490, 2015.

[12] G. Hernandez, G. LaMountain, and P. Closas. Privacy-preserving cooperative positioning. In *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, pages 2667–2675, 2020.

[13] A. Ibarrondo and A. Viand. Pyfhel: Python for homomorphic encryption libraries. In *Proceedings of the 9th on Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, pages 11–16, 2021.

[14] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione. A survey of enabling technologies for network localization, tracking, and navigation. *IEEE Communications Surveys & Tutorials*, 20(4):3607–3644, 2018.

[15] A. Mackey, P. Spachos, L. Song, and K. N. Plataniotis. Improving ble beacon proximity estimation accuracy through bayesian filtering. *IEEE Internet of Things Journal*, 7(4):3160–3169, 2020.

[16] Y. J. Morton, F. van Diggelen, J. J. Spilker Jr, B. W. Parkinson, S. Lo, and G. Gao. *Position, Navigation, and Timing Technologies in the 21st Century: Integrated Satellite Navigation, Sensor Systems, and Civil Applications*. John Wiley & Sons, 2021.

[17] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, D. Boneh, et al. Location privacy via private proximity testing. In *NDSS*, volume 11, 2011.

[18] E. Novak and Q. Li. Near-pri: Private, proximity based location sharing. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 37–45. IEEE, 2014.

[19] X. Peng, R. Chen, K. Yu, F. Ye, and W. Xue. An improved weighted k-nearest neighbor algorithm for indoor localization. *Electronics*, 9(12):2117, 2020.

[20] S. Shiraki and S. Shioda. Contact information-based indoor pedestrian localization using bluetooth low energy beacons. *IEEE Access*, 10:119863–119874, 2022.

[21] S. Subedi, H.-S. Gang, N. Y. Ko, S.-S. Hwang, and J.-Y. Pyun. Improving indoor fingerprinting positioning with affinity propagation clustering and weighted centroid fingerprint. *IEEE Access*, 7:31738–31750, 2019.

[22] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 24–43. Springer, 2010.

[23] B. Wang, Q. Chen, L. T. Yang, and H.-C. Chao. Indoor smartphone localization via fingerprint crowdsourcing: Challenges and approaches. *IEEE Wireless Communications*, 23(3):82–89, 2016.

[24] S. Xu, R. Chen, Y. Yu, G. Guo, and L. Huang. Locating smartphones indoors using built-in sensors and wi-fi ranging with an enhanced particle filter. *IEEE Access*, 7:95140–95153, 2019.

[25] R. K. Yadav, B. Bhattarai, H.-S. Gang, and J.-Y. Pyun. Trusted k nearest bayesian estimation for indoor positioning system. *IEEE Access*, 7:51484–51498, 2019.

[26] F. Yin, Y. Zhao, and F. Gunnarsson. Proximity report triggering threshold optimization for network-based indoor positioning. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 1061–1069. IEEE, 2015.

[27] Y. Yu, R. Chen, L. Chen, X. Zheng, D. Wu, W. Li, and Y. Wu. A novel 3-d indoor localization algorithm based on ble and multiple sensors. *IEEE Internet of Things Journal*, 8(11):9359–9372, 2021.