Liu, Shushu; Yan, Zheng

Efficient Privacy Protection Protocols for 5G Enabled Positioning in Industrial IoT

Published in:
IEEE Internet of Things Journal

DOI:
10.1109/JIOT.2022.3161148

Published: 01/10/2022

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Liu, S., & Yan, Z. (2022). Efficient Privacy Protection Protocols for 5G Enabled Positioning in Industrial IoT. *IEEE Internet of Things Journal*, *9*(19), 18527-18538. https://doi.org/10.1109/JIOT.2022.3161148

Efficient Privacy Protection Protocols for 5G Enabled Positioning in Industrial IoT

Shushu Liu, *Member, IEEE,* Zheng Yan, *Senior Member, IEEE*

***Abstract*—High-accuracy positioning has drawn huge attention with the potential in enhancing location-aware communications, intelligent transportation, and so on. The emergency of the fifth-generation (5G) technologies like device to device (D2D) communications, vehicle to vehicle (V2V) communications and crowdsourcing networks is expected to help achieve highly accurate positioning. By employing nearby mobile terminals to estimate position cooperatively, these technologies can improve positioning accuracy effectively, especially in indoor and urban areas. Despite the benefit, the potential information disclosure in these positioning systems threatens the engagement of public participants (also known as reference points). The location of the reference points and their distances to a target point is quite sensitive since they can be easily used to locate the reference points once exposed. Though existing solutions based on Paillier homomorphic encryption have been proposed to preserve the privacy of distance information. The sensitivity of reference points' locations is ignored. Additionally, the adoption of Paillier introduces a high computation cost, which is impractical in reality. To address the above problems, this paper proposes two efficient protocols, named Pub-pos and Pri-pos. By leveraging matrix concatenation and multiplication, these two protocols can disguise the original sensitive data, including both distance and location information, into a random matrix while keeping a positioning result intact. We analyze security strength, complexity and optimal variable selection of the proposed protocols. Numerous experiments verify that our proposed protocols have significant efficiency improvement in both system and individual levels compared with a Paillier based solution.**

***Index Terms*—Location Privacy Protection, D2D Positioning, D2D Communications, 5G, V2V, Cloud Computing.**

I. Introduction

THE industrial internet of things (IIoT) improves the industry productivity and efficiency significantly with the connectivity and smart automation enabled by Internet of Things (IoT) devices and cloud computing technology. As one of the central technical enablers in the emerging IIoT, high-accuracy positioning has drawn huge attention with the potential in enhancing location-aware communications, intelligent transportation, and so on [1]. Positioning is also a crucial function in location-based services (LBS) like Google Maps, Uber, Wolt, etc.

Owing to the crucial importance of positioning and urgent expectation on accurate solutions, the fifth-generation (5G) network aims to provide the ubiquitous positioning of below one-meter accuracy, especially in indoor and urban areas [2]. The emergence of new technology like device to device (D2D) communications [3], vehicle to everything (V2X) communications [4], crowdsourcing [5], and even unmanned aerial vehicle (UAV) [6] creates the possibility for this goal. With these technologies, two mobile terminals can communicate directly without going through the core network. For positioning, these technologies enable mobile terminals to cooperate for achieving accurate coordinates by allowing directly exchanging necessary data through the enabled communication links. These enabled communication links provide more Line-of-sight (LOS) communication and can reduce network delay in data exchange, thus achieving better positioning results.

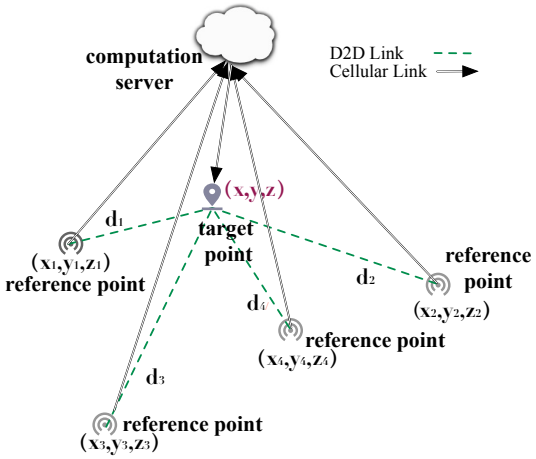


Fig. 1: Example of D2D enabled positioning

Figure 1 shows an example of D2D enabled positioning. It involves three types of participants: a target point, several reference points and a computation server. The target point is the device requests for positioning services, and the reference points are nearby devices ready to provide positioning assistance. The computation server is a center with computation power, which can be an edge point or a cloud platform. The target point communicates with the reference points nearby through enabled D2D links while the computation server communicates with both the target point and the reference points through cellular links. Each reference point is in possession of its coordinate (x, y, z) and can easily estimate its distance (d_i) to the target point by measuring the message travelling time from reference point to the target point. The computation server collects this information from at least four reference points and finds their intersection point by solving the quadratic equations based on Euclidean distance.

S. Liu is with the Department of Communications and Networking, Aalto University, Espoo, 02150 Finland e-mail: liu.shushu@aalto.fi

Z. Yan (corresponding author) is with the Department of Communications and Networking, Aalto University, Espoo, 02150 Finland and the State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an, 710071, China e-mail: zheng.yan@aalto.fi

Manuscript received December 30, 2021; revised xxx.

The intersection point is the estimated location of the target. The same model is adopted in vehicular ad hoc network (VANET) positioning systems [7], sensor network positioning systems [8] and so on. In VANET, the target point is normally a car and the reference points can be roadside units or other surrounding cars while the base station nearby can serve as the computation server. Extensive studies [9]–[11] have been devoted to improving the accuracy and efficiency of this positioning process.

Despite the benefits of these positioning systems, the risk of information disclosure threatens the engagement of their participants. The threat resides in three aspects: (1) The estimated coordinate of the target point. This results from the sensitivity of location as the possibility of inferring sensitive information related to the location owner. For example, a user's home address could be inferred if his location is in a residential area or his health status could be inferred if his location is a hospital for some specific disease treatment. (2) The coordinate of the reference point. It should be kept secret to prevent the potential information leakage of reference points just like the above discussion. However, the exception also exists when the reference points are public infrastructures with no privacy required such as base stations and roadside units in VANET. (3) The distance between the target point and reference points. This is treated as sensitive for two reasons. First, it risks the privacy of the target point as adversarial attackers can use it to locate the target point when combined with public reference points. Second, it risks the privacy of the reference point as a malicious target point could use it reversely to locate a private reference point based on the historical interaction data. Thus, distance information is sensitive and should be kept secret.

The challenge to provide a privacy-preserving positioning solution is to solve quadratic equations when multiple participants are involved. Solutions based on cryptographic tools have been proposed. Based on a one-pass authentication key agreement protocol, Pei et al. [7] implemented two secure and privacy-preserving 3D positioning schemes. Hussain and Koushanfar [12] proposed vehicle positioning solutions with Garbled Circuit and Beaver Micali Rogaway. However, these solutions are impractical owing to their high computational overheads. Shu et al. [13] solved the problem with a new perspective. They first formulated the triangulation into a least-square-error (LSE) estimation problem and then implemented its privacy-preserving computation with matrix multiplication and homomorphic encryption. However, this model is constructed in a pervasive environment where the target point is used as the computation center. While this paper considers an untrusted computation center, which is a general case for cloud or edge computing settings. Another solution is presented by Jiang et al. in [14]. Based on the homomorphic property of Paillier encryption, they implemented location calculation over protected distance. However, this solution is infeasible to the crowdsourcing environment when the reference points could be a private worker who requests to protect its location privacy. Besides, the overhead introduced by Paillier is prohibitive in practice.

To mitigate the privacy risk in these emerging positioning systems, two protocols, named Pub-pos and Pri-pos, with different privacy constraints are proposed in this paper. Pub-pos is a privacy-preserving positioning protocol that assumes the public availability of reference points' locations. It aims to prevent the privacy leakage of measured distance and compute the location of the target point. Pri-pos is designed for the scenario where the locations of reference points are also sensitive, needed to be protected. Our general idea is to implement privacy preservation by leveraging matrix concatenation and multiplication. Specifically, an encryption key that includes several random matrices is firstly generated and distributed by the target point. Based on the key, the reference point disguises its original data as random by matrix concatenation and multiplication operations before sending it to the computation server. With these random data, the computation server computes and sends an encrypted location back to the target point, where the location can be easily revealed with the encryption key. The designed protocols are useful in many use cases. For example, Pub-pos is helpful in privacy preservation for outdoor positioning based on base stations [15] or VANET positioning based on roadside units [16], where the locations of reference infrastructures are normally known. Pri-pos can be applied to scenarios where not only the distance but also the coordinates of references are quite sensitive and request protection, e.g., indoor positioning based on crowdsourcing workers or VANET positioning based on vehicles.

In general, the contributions of this paper can be summarized as follows.

- For preserving privacy in D2D based positioning, we design two protocols, Pub-pos and Pri-pos, based on matrix concatenation and multiplication. Compared with traditional cryptographic solutions, the proposed protocols present superiority with regard to high efficiency and low service latency.
- The correctness and security of the proposed protocols are theoretically proved.
- Extensive discussion is conducted on Pub-pos and Pri-pos in terms of security strength, complexity and the optimal selection of variables.
- The performance of proposed protocols are evaluated both theoretically and experimentally. Experimental results show the high efficiency of our proposed protocols compared with existing solutions.

The rest of the paper is structured as follows. We review the related work in Section II. An overview of three dimensional (3D) positioning and its threat model is presented in Section III. Depending on privacy protection demands, a privacy preservation protocol based on public reference points is presented in Section IV, followed by a privacy-preserving protocol based on private reference points in Section V. Analysis on security strength, complexity and selection of variables are presented in Section VI. Section VII presents our experimental results and comparison with related work. Finally, we summarize the paper in the last section.

II. Related Work

Privacy-preserving positioning aims to protect the private information of the participants involved in the positioning

process. A variety of privacy-preserving schemes are proposed for both indoor [17] and outdoor [18], [19]. For indoor positioning based on RSS, Konstantinidis et al. [20] proposed a scheme based on bloom filter and k-anonymity. A localization query is first mapped into a Bloom filter vector and further obfuscated with another $k-1$ vectors before sending it to a computation server. Although privacy can be protected under k-anonymity, it requires a trusted third party for obfuscation. Similar work is also presented by Sazdar et al. in [21] based on a bloom filter.

Also for indoor positioning, Li et al. [22] implemented a privacy-preserving solution for fingerprint-based localization. By leveraging the homomorphic property of the Paillier cryptosystem, computation servers can compute the Euclidean distance between a query and database in the ciphertext and respond with the nearest location. In this way, the privacy of the query can be protected. However, this solution is discovered with disclosure of the server's database when the client is not totally "honest" and is allowed to send fabricated queries to the server as spotted by Yang and Järvinen in [23]. They further enhanced the scheme with two solutions based on fully homomorphic encryption and garbled circuits separately. Even though, the heavy computation cost generated from cryptographic protocols is still a burden to prevent the scheme in practice.

Schauer, Dorfmeister and Wirth [24] proposed a method to prevent privacy leakage in indoor positioning. To prevent user privacy leakage in probe requests when accessing localization-based services, they adapted the communication protocol IEEE 802.11 used between an access point and a user device from active scan to passive scan, so that the user device only listens to periodical signals sent by access points. By abandoning active communication from mobile users to access points, this method suffers from positioning accuracy decrease.

Apart from indoor positioning, related work also appeared in VANET and image-based positioning. Based on a one-pass authentication key agreement protocol, Pei et al. [7] implemented two secure and privacy-preserving 3D positioning schemes. Hussain and Koushanfar [12] proposed vehicle positioning methods with garbled circuits and Beaver Micali Rogaway. A privacy preservation scheme for image-based localization was presented by Speciale et al. [25] to avoid the confidential information deduction from the collected 3D scene while allowing reliable camera pose estimation. This scheme lifts a map representation from a traditional 3D point cloud to a 3D line cloud.

Shu et al. [18] solved the problem with a new perspective. They first formulated the localization as least-square-error (LSE) estimation and then implemented its secure computation with matrix multiplication and homomorphic encryption. However, this model is constructed based on a pervasive environment where the target point is used as the computation center. While this paper considers an untrusted computation center which is a more general case for cloud or edge computing settings. Another solution is presented by Jiang et al. [14] based on Paillier encryption. It implemented location calculation over encrypted distance information with distance protection. However, their solution is only limited to scenarios when the location information of reference points are available. It is not feasible to D2D scenarios when the reference points could be a crowdsourcing worker who requires location privacy. Besides, the overhead introduced by Paillier is prohibitive in practice.

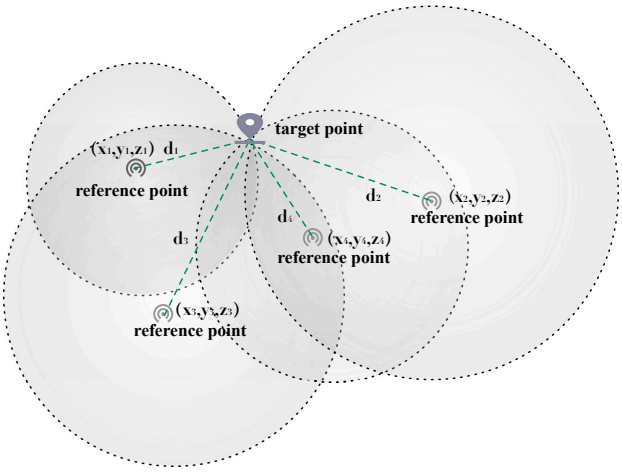


Fig. 2: Overview of trilateration based 3D positioning

III. Positioning and Threat Model

A. Overview of Trilateration based 3D Positioning

As the most representative positioning method for both indoor and outdoor, trilateration solves the problem by finding the intersection of spheres which are defined as a system of quadratic equations. Fig. 2 is an example of 3D positioning based on trilateration. It normally includes a target point and at least four reference points with known locations. We suppose that these reference points are not in the same plane and the target point is located in the range of these reference points. At least four spheres should be created to estimate the location of the target point. The sphere can be expressed as a quadratic equation according to Euclidean geometry, based on the coordinates of target point (x, y, z), reference points (x_i, y_i, z_i) and the distance d_i from reference points to the target point. The target point communicates with reference points nearby through enabled D2D links. During the communication, the reference point can easily estimate its distance (d_i) to the target point through message travelling time. All four spheres are created similarly. The location of the target point can be obtained by finding the intersection point of four spheres, which in theory is equivalent to solving the independent linear equations as shown in Equation 1.

$$\begin{cases} (x-x_1)^2+(y-y_1)^2+(z-z_1)^2 = {d_1}^2 \\ (x-x_2)^2+(y-y_2)^2+(z-z_2)^2 = {d_2}^2 \\ (x-x_3)^2+(y-y_3)^2+(z-z_3)^2 = {d_3}^2 \\ (x-x_4)^2+(y-y_4)^2+(z-z_4)^2 = {d_4}^2 \end{cases} \tag{1}$$

Through multinomial expansion, Equations 1 can be expressed as matrix computation as

$$AX = B, \tag{2}$$

where,

$$A = 2\begin{bmatrix}(x_1-x_2) & (y_1-y_2) & (z_1-z_2)\\ (x_1-x_3) & (y_1-y_3) & (z_1-z_3)\\ (x_1-x_4) & (y_1-y_4) & (z_1-z_4)\end{bmatrix}, X = \begin{bmatrix}x\\ y\\ z\end{bmatrix}$$

$$B = \begin{bmatrix}{d_2}^2-{x_2}^2-{y_2}^2-{z_2}^2-{d_1}^2+{x_1}^2+{y_1}^2+{z_1}^2\\ {d_3}^2-{x_3}^2-{y_3}^2-{z_3}^2-{d_1}^2+{x_1}^2+{y_1}^2+{z_1}^2\\ {d_4}^2-{x_4}^2-{y_4}^2-{z_4}^2-{d_1}^2+{x_1}^2+{y_1}^2+{z_1}^2\end{bmatrix}$$

Thus, the intersection point in equation 1 can be revealed by one matrix inverse and one matrix multiplication as

$$X = A^{-1}B. \tag{3}$$

B. Threat Model

As presented in Fig. 1, the whole positioning system is composed of the target point, the reference points and the computation server. The reference points are always listening to the target point and measuring their distances once requested. Afterwards, all the reference points send their coordinates (x_i, y_i, z_i) and the measured distance d_i to the computation server. Based on these collected data from reference points, the computation server can determine the coordinates of the target point by applying Equation 3 directly.

We assume that all participants of the positioning system are honest but curious. An honest but curious participant may execute the proposed protocol as designed but is also eager in collecting related information of others that could be leaked during the positioning process. An observation of the definition of A and B in Equation 2 reveals that normally calculating X requires all reference points to disclose their coordinates (x_i, y_i, z_i) and distance d_i to the computation server. Since the sensitivity of this information, we need to provide a solution to prevent its leakage while allowing the positioning process to run smoothly.

Besides, we also suppose that the communications between participants are reliable, based on secure authentication or encryption techniques, so that privacy leakage does not come from the communications channel. In addition, we assume three types of participants are all independent parties and there is no collusion between any two parties. The target point does not collaborate with either the reference points or the computation server since this is against its privacy. Since the reference points are normally independent crowdsourcing workers, their collusion with the computation server may reveal their private information and also negatively impact their reputations. Thus, collusion among the three types of system participants is not considered in our research. Besides, we do not consider any active attack of participants, such as false location information injection of the reference points, manipulation of the computation, or modification of results, to mislead or cheat the target. All the above attacks are valid to the positioning system and could be detected through data analysis and machine learning-based methods [26]. However, it is out of the scope of this paper. Here we mainly focus on preventing privacy leakage in a normal positioning computation.

IV. Pub-pos: Privacy Preserving Positioning Based on Public Reference Points

Algorithm 1: Privacy-preserving Positioning Based On Public Reference Points

Result: U with coordinates X
Private Inputs: R_i with distance d_i
Public Inputs: R_i with coordinate (x_i, y_i, z_i) ;
Step1 (Target Point U):
1.1: generate four $3 \times l$ matrices RB_1, RB_2, RB_3, RB_4 with k-bits random value and $RB_1 = RB_2 + RB_3 + RB_4$;
1.2: generate a $(l+1) \times 1$ matrix K^B with k-bits random value ;
1.3: distribute $(RB_1, K^B), (RB_2, K^B)$, $(RB_3, K^B), (RB_4, K^B)$ to reference points R_1, R_2, R_3, R_4 randomly;
Step2 (Reference Points R_1, R_2, R_3, R_4):
2.1: each accept key pair (RB_i, K^B) respectively;
2.2: each compose a matrix with its distance d as

$$B_1 = \begin{bmatrix}d_1^2\\ d_1^2\\ d_1^2\end{bmatrix}, B_2 = \begin{bmatrix}d_2^2\\ 0\\ 0\end{bmatrix}, B_3 = \begin{bmatrix}0\\ d_3^2\\ 0\end{bmatrix}, B_4 = \begin{bmatrix}0\\ 0\\ d_4^2\end{bmatrix}$$

2.3: each merge B and RB by row concatenation, as $D = \begin{bmatrix}B & RB\end{bmatrix}$;
2.4: each encrypt D with K^B as $E = DK^B$;
2.5: each sends it E denoted as E_1, E_2, E_3, E_4 to the computation server S;
Step3 (Computation Server S):
3.1: accept E_1, E_2, E_3, E_4 ;
3.2: compose A and B_5 based on public coordinates as

$$A = 2\begin{bmatrix}(x_1-x_2) & (y_1-y_2) & (z_1-z_2)\\ (x_1-x_3) & (y_1-y_3) & (z_1-z_3)\\ (x_1-x_4) & (y_1-y_4) & (z_1-z_4)\end{bmatrix}$$

$$B_5 = \begin{bmatrix}-{x_2}^2-{y_2}^2-{z_2}^2+{x_1}^2+{y_1}^2+{z_1}^2\\ -{x_3}^2-{y_3}^2-{z_3}^2+{x_1}^2+{y_1}^2+{z_1}^2\\ -{x_4}^2-{y_4}^2-{z_4}^2+{x_1}^2+{y_1}^2+{z_1}^2\end{bmatrix}$$

3.3: compute $C = A^{-1}(-E_1 + E_2 + E_3 + E_4), F = A^{-1}B_5$;
3.4: send C and F to U;
Step4 (Target Point U):
4.1: accept C and F;
4.2: reveal the location as $X = Cm^{-1} + F$ where $m = K_1^B$;

Pub-pos assumes that the coordinates of each reference point are public. They obtain their coordinates from GPS and publicise them for assisting positioning services. However, the distance information held by these reference points is sensitive and required to be kept secret from the computation server. The computation server is responsible for collecting the positioning-related data from each reference point and sending its computation result to the target point. Notably, such a localization model is popularly used in outdoor positioning

based on base stations and VANET positioning based on roadside units.

The protocol to achieve the above requirements is presented in Algorithm 1. Target point U starts the protocol by generating four $3 \times l$ matrices RB_1, RB_2, RB_3, RB_4 and a $l+1 \times 1$ matrix K^B. It is required that $1 \leq l$ so that random matrix RB is not empty. Also, $13l+1 \leq 2^k$ so that there are enough random values for selection. Further discussion of the selection of parameters can be found in Section VI. It is also required that $RB_1 = RB_2 + RB_3 + RB_4$ and all random values in the matrix are no longer than k-bits. Herein, the selection of random values distribution is flexible. We generate random values following a uniform distribution, but other distributions like Gaussian or Laplace can also be applied since the privacy of our protocol can be preserved with no dependence on the distribution of the random values. U then distributes (RB_1, K^B), (RB_2, K^B) , (RB_3, K^B) and (RB_4, K^B) randomly to four reference points (R_1, R_2, R_3, R_4) through secure channels. The reference point R_i accepts RB_i and K^B. At the same time, it composes its distance information d_i into a 3×1 matrix B_i, as shown in step 2.2 of Algorithm 1. To randomize the distance matrix B_i, R_i first merges it with RB_i by row concatenation and further encrypts the merged matrix with random matrix K^B. The encrypted matrix denoted as E_i can then be sent to the computation server S. At S, A^{-1} and B_5 are first initialized according to the public knowledge of $\{(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4)\}$, as shown in step 3.2 of Algorithm 1. Once accepting the encrypted matrices from all reference points, S integrates the obtained matrix B_1, B_2, B_3, B_4 with A^{-1} as $C = A^{-1}(-B_1 + B_2 + B_3 + B_4)$. Meanwhile, it computes $F = A^{-1}B_5$. Both C and F are sent back to U once completed. Based on C and F, U reveals its location easily with $X = Cm^{-1} + F$ where $m = K_1^B$, as shown in step 3.2 of Algorithm 1.

A. Correctness Analysis

The correctness analysis can be derived as follow equation by substituting the formulas.

$$\begin{aligned} X &= Cm^{-1} + F \\ &= A^{-1}(-E_1 + E_2 + E_3 + E_4)m^{-1} + A^{-1}B_5 \\ &= A^{-1}(-D_1 + D_2 + D_3 + D_4)K^B m^{-1} + A^{-1}B_5 \end{aligned} \tag{4}$$

Firstly, we prove that

$$(-D_1 + D_2 + D_3 + D_4)K^B m^{-1} = -B_1 + B_2 + B_3 + B_4.$$

Since $D_i = \begin{bmatrix} B_i & RB_i \end{bmatrix}$ which is

$$\begin{bmatrix} d_i^2/0 & r_{11}^i & r_{12}^i & \cdots & r_{1l}^i \\ d_i^2/0 & r_{21}^i & r_{22}^i & \cdots & r_{2l}^i \\ d_i^2/0 & r_{31}^i & r_{32}^i & \cdots & r_{3l}^i \end{bmatrix}$$

and $RB_1 = RB_2 + RB_3 + RB_4$. It is easy to derive that $-D_1 + D_2 + D_3 + D_4$ is

$$\begin{bmatrix} {d_2}^2 - {d_1}^2 & 0 & 0 & \cdots & 0 \\ {d_3}^2 - {d_1}^2 & 0 & 0 & \cdots & 0 \\ {d_4}^2 - {d_1}^2 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Let $K^B = \begin{bmatrix} k_1 & k_2 & \cdots & k_l & k_{l+1} \end{bmatrix}^T$, we can have

$$(-D_1 + D_2 + D_3 + D_4)K^B = \begin{bmatrix} ({d_2}^2 - {d_1}^2)k_1 \\ ({d_3}^2 - {d_1}^2)k_1 \\ ({d_4}^2 - {d_1}^2)k_1 \end{bmatrix}$$

Also since $m = K_1^B = k_1$, and $(-D_1 + D_2 + D_3 + D_4)K^B m^{-1}$ is

$$\begin{bmatrix} {d_2}^2 - {d_1}^2 \\ {d_3}^2 - {d_1}^2 \\ {d_4}^2 - {d_1}^2 \end{bmatrix}$$

Thus, $(-D_1 + D_2 + D_3 + D_4)K^B m^{-1} = -B_1 + B_2 + B_3 + B_4$ is proved. Based on the proof, the following equation can be derived from Equation 4

$$\begin{aligned} X &= A^{-1}(-B_1 + B_2 + B_3 + B_4) + A^{-1}B_5 \\ &= A^{-1}(-B_1 + B_2 + B_3 + B_4 + B_5) \end{aligned}$$

According to the definition of B_1, B_2, B_3, B_4, B_5, we have $B = -B_1 + B_2 + B_3 + B_4 + B_5$ as

$$\begin{bmatrix} {d_2}^2 - {x_2}^2 - {y_2}^2 - {z_2}^2 - {d_1}^2 + {x_1}^2 + {y_1}^2 + {z_1}^2 \\ {d_3}^2 - {x_3}^2 - {y_3}^2 - {z_3}^2 - {d_1}^2 + {x_1}^2 + {y_1}^2 + {z_1}^2 \\ {d_4}^2 - {x_4}^2 - {y_4}^2 - {z_4}^2 - {d_1}^2 + {x_1}^2 + {y_1}^2 + {z_1}^2 \end{bmatrix}$$

Thus,

$$X = A^{-1}B$$

Thus, the correctness of the protocol can be proved.

B. Privacy Analysis

This part proves that the protocol presented in Algorithm 1 can preserve the privacy of (1) distance d_i of each reference point; (2) estimated coordinate X of the target point. The theorem and proof are presented below.

Theorem 1. *For all reference points, the distance d_i to the target point is protected from computation server S.*

For the reference point R_i, it accepts random matrices K^B and RB from target point U and uses them to encrypt its distance information d_i, as shown in steps 2.3 and 2.4 of Algorithm 1. The computed matrix E_i is then sent to the computation server S. We need to prove that S is unable to infer any information about d_i from either E_i or its combination.

To have a clear picture, we present D_i and K^B as

$$D_i = \begin{bmatrix} d_1^i & r_{11}^i & r_{12}^i & \cdots & r_{1l}^i \\ d_2^i & r_{21}^i & r_{22}^i & \cdots & r_{2l}^i \\ d_3^i & r_{31}^i & r_{32}^i & \cdots & r_{3l}^i \end{bmatrix}$$

$$K^B = \begin{bmatrix} k_1 & k_2 & \cdots & k_l & k_{l+1} \end{bmatrix}^T$$

where d^i is the distance information, r and k are k-bits random value. Based on D_i and K^B, we express E_i as

$$E_i = \begin{bmatrix} d_i^2 k_1/0 + r_{11}^i k_2 + r_{12}^i k_3 + \cdots + r_{1l}^i k_{l+1} \\ d_i^2 k_1/0 + r_{21}^i k_2 + r_{22}^i k_3 + \cdots + r_{2l}^i k_{l+1} \\ d_i^2 k_1/0 + r_{31}^i k_2 + r_{32}^i k_3 + \cdots + r_{3l}^i k_{l+1} \end{bmatrix}$$

Observing E_i, it contains $4l+2$ variables. It is impossible to reverse either d, r or k through matrix decomposition since

the insufficient equations compared with variables. Thus, it is impossible for S to infer information about R_i from single E_i. Another information come from the fact that $RB_1 = RB_2 + RB_3 + RB_4$. It can be used to remove the random r with

$$-E_1 + E_2 + E_3 + E_4 = \begin{bmatrix}(d_2^2 - d_1^2)k_1\\(d_3^2 - d_1^2)k_1\\(d_4^2 - d_1^2)k_1\end{bmatrix}$$

Randomized by the random k_1, both d and the difference between d can be disguised. Thus, we can conclude that computation server S cannot infer any information about reference points.

Theorem 2. *For target point U, the estimated coordinate X is protected from computation server S.*

For target point U, the information leakage also resides in computation server S. The risk is that S can infer X by reversing m, which is K_1^B from the observations. As we have analyzed in Theorem 1, the data hosted by S during the computation include E_1, E_2, E_3 and E_4. Based on the first column of E_1, E_2, E_3 and E_4, there are four equations with five variables. It is not enough to derive m from this quadratic equation. Thus, S is unable to reverse the coordinate X.

V. Pri-pos: Privacy Preserving Positioning Based on Private Reference Points

This section solves the positioning with private reference points. These reference points are location-sensitive entities who are unwilling to share their locations even to the computation server. The workflow is similar as presented in the previous section, but the detail is revised accordingly. Notably, such a privacy-preserving model is useful in positioning with location-sensitive reference points, like indoor positioning based on crowdsourcing worker, V2V assisted positioning for autonomous driving and so on. Compared with Pub-pos, the difference in this protocol is that the locations of reference points are private, which makes it challenging to calculate matrix A^{-1} and B_5.

The proposed protocol is presented in Algorithm 2. Apart from the random matrix K^B, target point U also generates random matrices RA, K^A, which will be used for the encryption of matrix A. To align with A, the dimension of RA is $3 \times s$ and K_A is $(s+3) \times 3$. For the parameters, it shares the same requirement for l as defined in Alg. 1. Besides, it is required that $14s + 6 \leq 2^k$ and $1 \leq s$ as the space requirement for random value and the random matrix. Further discussion of the selection of parameters can be found in Section VI. For the reference points, the matrix A and B are composed based on their inputs locally, as shown in step 2.2 of Algorithm 2. The same operations are conducted to encrypt A and B before sending them to the computation server S. They merge A and B with RA and RB by row concatenation first and then multiply with K^A and K^B, as shown from step 2.3 to step 2.6 in Algorithm 2. At the side of the computation server, it combines all received matrix into C through matrix addition, matrix inversion and matrix multiplication operations, as shown in step 3.2 of Algorithm 2. In the end, U reveals X with C and key K^A and K^B, as shown in step 4.2 of Algorithm 2.

A. Correctness Analysis

The correctness analysis can be derived through equation by substituting the formulas.

$$\begin{aligned}X &= HCm^{-1}\\ &= H(F_1 - F_2 - F_3 - F_4)^{-1}(-G_1 + G_2 + G_3 + G_4)m^{-1}\\ &= H(D_1K^A - D_2K^A - D_3K^A - D_4K^A)^{-1}\\ &(-E_1K^B + E_2K^B + E_3K^B + E_4K^B)m^{-1}\\ &= H((D_1 - D_2 - D_3 - D_4)K^A)^{-1}\\ &((-E_1 + E_2 + E_3 + E_4)K^B)m^{-1}\end{aligned} \tag{5}$$

Firstly, we prove the correctness of

$$(A_1 - A_2 - A_3 - A_4)H = (D_1 - D_2 - D_3 - D_4)K^A.$$

Since $D_i = \begin{bmatrix}A_i & RA_i\end{bmatrix}$ and $RA_1 = RA_2 + RA_3 + RA_4$, it is easy to get $D_1 - D_2 - D_3 - D_4$ as

$$2\begin{bmatrix}x_1 - x_2 & y_1 - y_2 & z_1 - z_2 & 0 & \cdots & 0\\ x_1 - x_3 & y_1 - y_3 & z_1 - z_3 & 0 & \cdots & 0\\ x_1 - x_4 & y_1 - y_4 & z_1 - z_4 & 0 & \cdots & 0\end{bmatrix}$$

Let

$$K^A = \begin{bmatrix}k_{11} & k_{12} & \cdots & k_{1s} & \cdots & k_{1(s+3)}\\ k_{21} & k_{22} & \cdots & k_{2s} & \cdots & k_{2(s+3)}\end{bmatrix}^T$$

Thus, $(D_1 - D_2 - D_3 - D_4)K^A$ is

$$2\begin{bmatrix}a_1k_{11} + b_1k_{12} + c_1k_{13} & a_1k_{21} + b_1k_{22} + c_1k_{23}\\ a_2k_{11} + b_2k_{12} + c_2k_{13} & a_2k_{21} + b_2k_{22} + c_2k_{23}\\ a_3k_{11} + b_3k_{12} + c_3k_{13} & a_3k_{21} + b_3k_{22} + c_3k_{23}\end{bmatrix}$$

where $a_i = (x_1 - x_{i+1}), b_i = (y_1 - y_{i+1})$ and $c_i = (z_1 - z_{i+1})$.

And since $(A_1 - A_2 - A_3 - A_4)H$ is

$$2\begin{bmatrix}a_1k_{11} + b_1k_{12} + c_1k_{13} & a_1k_{21} + b_1k_{22} + c_1k_{23}\\ a_2k_{11} + b_2k_{12} + c_2k_{13} & a_2k_{21} + b_2k_{22} + c_2k_{23}\\ a_3k_{11} + b_3k_{12} + c_3k_{13} & a_3k_{21} + b_3k_{22} + c_3k_{23}\end{bmatrix}$$

when

$$H = \begin{bmatrix}k_{11} & k_{21}\\ k_{12} & k_{22}\\ k_{13} & k_{23}\end{bmatrix}.$$

Thus, the correctness of $(A_1 - A_2 - A_3 - A_4)H = (D_1 - D_2 - D_3 - D_4)K^A$ can be proved.

Secondly, we prove the correctness of

$$-B_1 + B_2 + B_3 + B_4 = (-E_1 + E_2 + E_3 + E_4)K^B m^{-1}.$$

Since $E_i = \begin{bmatrix}B_i & RB_i\end{bmatrix}$ and $RB_1 = RB_2 + RB_3 + RB_4$, it is easy to get $-E_1 + E_2 + E_3 + E_4$ as

$$\begin{bmatrix}d_2^2 - x_2^2 - y_2^2 - z_2^2 - (d_1^2 - x_1^2 - y_1^2 - z_1^2) & 0 & \cdots\\ d_3^2 - x_3^2 - y_3^2 - z_3^2 - (d_1^2 - x_1^2 - y_1^2 - z_1^2) & 0 & \cdots\\ d_4^2 - x_4^2 - y_4^2 - z_4^2 - (d_1^2 - x_1^2 - y_1^2 - z_1^2) & 0 & \cdots\end{bmatrix}$$

Algorithm 2: Privacy-preserving Positioning Based On Private Reference Points

Result: U with coordinates X

Private Inputs: R_i with coordinate (x_i, y_i, z_i) and distance d_i ;

Step1 (Target Point U):

1.1: generate four $3 \times l$ matrices RB_1, RB_2, RB_3, RB_4 with k-bits random value and $RB_1 = RB_2 + RB_3 + RB_4$;

1.2: generate a $(l+1) \times 1$ matrix K^B with k-bits random value ;

1.3: generate four $3 \times s$ matrices RA_1, RA_2, RA_3, RA_4 with k-bits random value and $RA_1 = RA_2 + RA_3 + RA_4$;

1.4: generate a $(s+3) \times 2$ matrix K^A with k-bits random value ;

1.4: distribute (RA_1, RB_1, K^A, K^B), (RA_2, RB_2, K^A, K^B), (RA_3, RB_3, K^A, K^B) randomly to P_1, P_2, P_3 ;

Step2 (Reference Points R_1, R_2, R_3, R_4):

2.1: each accepts (RA, RB, K^A, K^B) respectively;

2.2: each composes matrices A and B with

$$A_1 = 2\begin{bmatrix} x_1 & y_1 & z_1 \\ x_1 & y_1 & z_1 \\ x_1 & y_1 & z_1 \end{bmatrix}, B_1 = \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 - z_1^2 \\ d_1^2 - x_1^2 - y_1^2 - z_1^2 \\ d_1^2 - x_1^2 - y_1^2 - z_1^2 \end{bmatrix}$$

$$A_2 = 2\begin{bmatrix} x_2 & y_2 & z_2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B_2 = \begin{bmatrix} d_2^2 - x_2^2 - y_2^2 - z_2^2 \\ 0 \\ 0 \end{bmatrix}$$

$$A_3 = 2\begin{bmatrix} 0 & 0 & 0 \\ x_3 & y_3 & z_3 \\ 0 & 0 & 0 \end{bmatrix}, B_3 = \begin{bmatrix} 0 \\ d_3^2 - x_3^2 - y_3^2 - z_3^2 \\ 0 \end{bmatrix}$$

$$A_4 = 2\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ x_4 & y_4 & z_4 \end{bmatrix}, B_4 = \begin{bmatrix} 0 \\ 0 \\ d_4^2 - x_4^2 - y_4^2 - z_4^2 \end{bmatrix}$$

where d_i is the distance, (x_i, y_i, z_i) is the coordinates;

2.3: each merges A and RA into a new matrix D as $D_i = \begin{bmatrix} A_i & RA_i \end{bmatrix}$;

2.4: each merges B and RB into a new matrix E as $E_i = \begin{bmatrix} B_i & RB_i \end{bmatrix}$;

2.5: each encrypts D with K^A as $F = DK^A$;

2.6: each encrypts E with K^B as $G = EK^B$;

2.7: each sends F and G to computation server S;

Step3 (Computation Server S):

3.1: accept F_1, F_2, F_3, F_4 and G_1, G_2, G_3, G_4 ;

3.2: compute $C = (F_1 - F_2 - F_3 - F_4)^{-1}(-G_1 + G_2 + G_3 + G_4)$;

3.4: send C to U;

Step4 (Target Point U):

4.1: accept C ;

4.2: reveal the coordinates by $X = HCm^{-1}$ where $H = \begin{bmatrix} K_{11}^A & K_{12}^A \\ K_{21}^A & K_{22}^A \\ K_{31}^A & K_{32}^A \end{bmatrix}$ and $m = K_1^B$;

Let $K^B = \begin{bmatrix} k_1 & k_2 & \cdots & k_l & k_{l+1} \end{bmatrix}^T$, linearlize the following matrix, we have $(-E_1 + E_2 + E_3 + E_4)K^B$ as

$$\begin{bmatrix} (d_2^2 - x_2^2 - y_2^2 - z_2^2 - (d_1^2 - x_1^2 - y_1^2 - z_1^2))k_1 \\ (d_3^2 - x_3^2 - y_3^2 - z_3^2 - (d_1^2 - x_1^2 - y_1^2 - z_1^2))k_1 \\ (d_4^2 - x_4^2 - y_4^2 - z_4^2 - (d_1^2 - x_1^2 - y_1^2 - z_1^2))k_1 \end{bmatrix}$$

When $m = k_1$, the correctness of $-B_1 + B_2 + B_3 + B_4 = (-E_1 + E_2 + E_3 + E_4)K^B m^{-1}$ is proved.

By substituting the above equations into formula 5, we can get

$$\begin{aligned} X &= H((A_1 - A_2 - A_3 - A_4)H)^{-1}(-B_1 + B_2 + B_3 + B_4) \\ &= HH^{-1}(A_1 - A_2 - A_3 - A_4)^{-1}(-B_1 + B_2 + B_3 + B_4) \\ &= (A_1 - A_2 - A_3 - A_4)^{-1}(-B_1 + B_2 + B_3 + B_4) \\ &= A^{-1}B \end{aligned}$$

Thus, the correctness of the protocol can be proved.

B. Privacy Analysis

The privacy analysis of Algorithm 2 includes protecting the privacy of three aspects: (1) each reference point's distance to the target point; (2) coordinate of each reference point; (3) estimated coordinate of the target point. The theorems and proofs are presented below.

Theorem 3. *For each reference points R, its distance to the target point d_i is protected from the computation server S.*

According to step 2 of Algorithm 2, the distance information d_i of each reference point is first composed into matrix B_i in the format of $d_i^2 - x_i^2 - y_i^2 - z_i^2$. It is then merged with random matrix RB_i and encrypted by K^B before sending to the computation server. Thus, the knowledge obtained by the computation server is $G_i = \begin{bmatrix} B_i & RB_i \end{bmatrix} K^B$. In detail, we can linearlize G_i as

$$\begin{bmatrix} B_1^i k_1 + r_{11}^i k_2 + r_{12}^i k_3 + \cdots + r_{1l}^i k_{l+1} \\ B_2^i k_1 + r_{21}^i k_2 + r_{22}^i k_3 + \cdots + r_{2l}^i k_{l+1} \\ B_3^i k_1 + r_{31}^i k_2 + r_{32}^i k_3 + \cdots + r_{3l}^i k_{l+1} \end{bmatrix}$$

Similarly, it is impossible to reverse B_i, RB_i or K^B through any G_i since the insufficient knowledge about variables. Also, with the information of $RB_1 = RB_2 + RB_3 + RB_4$, it can remove the random r and reveals $-G_1 + G_2 + G_3 + G_4$ as

$$\begin{bmatrix} (d_2^2 - x_2^2 - y_2^2 - z_2^2 - (d_1^2 - x_1^2 - y_1^2 - z_1^2))k_1 \\ (d_3^2 - x_3^2 - y_3^2 - z_3^2 - (d_1^2 - x_1^2 - y_1^2 - z_1^2))k_1 \\ (d_4^2 - x_4^2 - y_4^2 - z_4^2 - (d_1^2 - x_1^2 - y_1^2 - z_1^2))k_1 \end{bmatrix}$$

However, randomized by k_1, both d_i and (x_i, y_i, z_i) of reference points can be disguised. Thus, we can conclude that S cannot infer any information of d_i.

Theorem 4. *For all reference points R, its coordinate (x_i, y_i, z_i) is protected from computation server S.*

The coordinate of reference point is involved in both the composition of matrix A and B. According to analysis of theorem 3, the privacy of B is preserved, thus the coordinate in B is protected. We only need to analyze the information leakage in matrix A. As we can see in steps 2.2, 2.3 and 2.5 of Algorithm 2, A_i is merged with RA_i and encrypted by

TABLE I: COMPLEXITY ANALYSIS

	Computation Cost			Communication Cost		
	U	R	S	U $\rightarrow$ R	R $\rightarrow$ S	S $\rightarrow$ U
Pub-pos	2	$3(l+1)$	18	$(4l+1)\times k$	$3\times 3k$	$6\times 3k$
Pri-pos	9	$(s+3)\times 6+(l+1)\times 3$	12	$(4l+5s+7)\times k$	$9\times 3k$	$2\times 6k$
Note: s and l are the sizes of random matrices RA and RB, respectively; k is the bit-length of random value						

TABLE II: SECURITY STRENGTH AND RECOMMENDED KEY SIZES

Security Strength	Paillier	Pub-pos	Pri-pos
Low (112-bits)	1024	$l=36, k=5$	$l=36, s=24, k=5$
Medium (128-bits)	2048	$l=48, k=9$	$l=48, s=37, k=9$
High (192-bits)	3072	$l=58, k=12$	$l=58, s=48, k=12$
Note: the variables are decided when the cost function is defined over the whole system and the percentage of both computation and communication cost are 50% and 50%, respectively			

random matrix K_A as $F_i = \begin{bmatrix} A_i & RA_i \end{bmatrix} K_A$ before sending to computation server S. Similarly, it is impossible to reverse A_i, RA_i or K^A through any F_i since insufficient number of equations compared with the number of variables. Also, with the information of $RA_1 = RA_2 + RA_3 + RA_4$, it reveals $F_1 - F_2 - F_3 - F_4$ as

$$\begin{bmatrix} a_1k_{11}+b_1k_{12}+c_1k_{13} & a_1k_{21}+b_1k_{22}+c_1k_{23} \\ a_2k_{11}+b_2k_{12}+c_2k_{13} & a_2k_{21}+b_2k_{22}+c_2k_{23} \\ a_3k_{11}+b_3k_{12}+c_3k_{13} & a_3k_{21}+b_3k_{22}+c_3k_{23} \end{bmatrix}$$

where $a_i = (x_1 - x_{i+1}), b_i = (y_1 - y_{i+1}), c_i = (z_1 - z_{i+1})$. However, owing to the randomization of K^A, the coordinates of reference points are kept secret from S.

Theorem 5. *For target point U, its coordinate X is protected from computation server S.*

According to the computation of target point U, H and m are required to reveal X, where H is the first two columns of K^A and m is the first value of K^B. Based on the theorem 3 and 4, K^A and K^B are kept secret from computation server S. Thus, X can be decrypted by target point U only.

VI. Discussion of Security Strength, Complexity and Variable selections

A. Security Strength

Security strength is associated with the number of operations required to break an algorithm or a system [27]. NIST[1] suggests to specify the security strength in bits and divide it into five levels as $\{80, 112, 128, 192, 256\}$. The bits represent the operations needed. For example, 2^{80} operations are needed to break an algorithm with 80-bits security strength, which translates to a few days on an average computer. Normally, the longer bits preserves a more secure algorithm. According to the computation power nowadays, NIST recommends 112 bits as the minimum. Throughout our experiments, we consider the security strength with three levels, denoted as "short (112-bits)", "medium (128-bits)" and "long (192-bits)". Under these settings, we can ignore the information leakage under brute force attacks. The security strength of our proposed protocols is derived as follows.

For Pub-pos, the security strength relies on the difficulty in breaking E. To infer D and K^B through E by brute force, it would take $2^{k\times(3l+1+l+1)}$ combinations and $3l$ multiplication in each combination. So the total multiplication for brute force attack would be $2^{k\times(4l+2)} \times 3l$, which is corresponding to $2^{k\times(4l+2)} \times 3l \times k^2$ xor operations. According to [27], the computation should be between $2^{1024}, 2^{2048}, 2^{3072}$ to achieve 112-bits, 128-bits and 192-bits security strength respectively. At the same time, it is required that $2^k \geq 13l+1$ since the space of a random value should be bigger than the number of required random values and $l \geq 1$ so that random matrix RB is not empty. Thus, it can be constrained into the following equations:

$$\begin{cases} 2^{t_1} \leq 2^{k\times(4l+2)} \times 3l \times k^2 \leq 2^{t_2} \\ 13l+1 \leq 2^k \\ 1 \leq l \end{cases} \tag{6}$$

with t_1 and t_2 are the operation required at each security strength level. For example, t_1 and t_2 are 1024 and 2048 for 112-bits security strength.

For Pri-pos, the security strength relies on the difficulty in breaking F and G. Since G is encrypted with the same process as Pub-pos, thus, the risk for breaking G is the same as analyzed above. To infer D and K^A through F, it would require $2^{k\times((3s+1+(s+3)\times 2)}$ combinations and $6(s+3)$ multiplication in each combination. Besides, $14s+6 \leq 2^k$ and $1 \leq s$ are the requirements for the space of random value and random matrix. Thus, the security strength of Pri-pos is equal to the following equation:

$$\begin{cases} 2^{t_1} \leq 2^{k\times(4l+2)} \times 3l \times k^2 \leq 2^{t_2} \\ 2^{t_1} \leq 2^{k\times(5s+7)} \times 6(s+3) \times k^2 \leq 2^{t_2} \\ 13l+1 \leq 2^k; 14s+6 \leq 2^k \\ 1 \leq l; 1 \leq s \end{cases} \tag{7}$$

with t_1 and t_2 are the operation required at each security strength level.

It is noteworthy that the security strength of our proposed protocols depends on the selection of its variables. However, to decide the optimal variables in practice, we also need to consider the computation and communication cost generated, which will be further discussed later.

B. Complexity Analysis

The complexity analysis of proposed protocols are listed in Table I. It includes both the computation and communication

[1]https://www.nist.gov/

costs of the two proposed protocols. U, R, S denotes the three participants in the positioning system and $\rightarrow$ denotes the data transformation from one to another. The computation cost is measured by the number of the most time-consuming operation, multiplication. The communication cost is measured by the bit-length of transferred values with respect to the size of the random matrix RA and RB.

In Pub-pos, U generates and distributes a $(l+1)\times 1$ matrix K^B and four $3\times l$ matrices RB_1, RB_2, RB_3, thus, the communication cost from U to R is $(4l+1)k$, where k is the bit-length of random value. Each R locally encrypts its distance matrix with K^B and sends it to S, thus, the computation for each R is the multiplication between a $3\times(l+1)$ matrix and a $(l+1)\times 1$ matrix, which results in $3(l+1)$ multiplications. After the multiplication, the value of E is $d^2k_1 + r_{11}^i k_2 + \cdots + r_{1l}^i k_{l+1}$, which can be presented in $3k$ bits. Thus, the communication cost from R to S is the size of matrix E, which equals $3\times 3k$. After collecting all the encrypted matrices from R, S executes matrix inverse computation over A and generates 6 multiplication operations, and computes C and F with 2 multiplication between a 3×3 matrix and a 3×1 matrix which in total is 18 multiplications. The two computed matrix C and F are sent to U later with six values and each value is no bigger than $3k$. In the end, U reveals X through matrix multiplication with 2 multiplication operations.

In Pri-pos, U starts the protocol by generating random matrices RA, RB, K^A, K^B and sends them to R, which corresponds to $4l+1+5s+6$ k-bits random values, thus, the communication cost from U to R is $(4l+5s+7)k$. To encrypt information matrices A and B, each R needs to execute $F = DK^A$ and $G = EK^B$. According to the size of each matrix, we can conclude the computation cost of R is $(s+3)\times 6 + (l+1)\times 3$ and the bit-length of F and G after computation is $3k$. Thus, the communication cost from R to S is $9\times 3k$. At the side of S, it first composes A and B through two matrix addition respectively. Based on the result, it computes C through one matrix inverse operation with 6 multiplication and one matrix multiplication between a 2×3 matrix and a 3×1 matrix with 6 multiplications. The computed C is sent to U with communication cost $2\times 6k$, where $6k$ is the bit-length of the value in C. In the end, U reveals its coordinates with HCm^{-1}, which is correspond to 9 multiplication when the size of H and C are 3×2 and 2×1.

From Table I, we can see that the computation cost of U and S are fixed, while the computation cost of P and all the communication costs between each participant are influenced by the selection of l, s and k.

C. Discussion of Optimal Variables

The criterion of variable selection is to generate least overhead while preserving the security strength. To make the process easy to understand, we formulate the overhead into a cost function which is defined as

$$F_{cost} = \alpha \times computation + \beta \times communication,$$

where α, β are the percentage of each cost and $\alpha + \beta = 1$. It is noteworthy that the computation and communication cost can be from either a single participant or from the whole system. Let's take Pub-pos as an example, we can define the cost function for U as $\alpha 2 + \beta((3l+1)\times k + 12k)$ or for the whole system as $\alpha(2+2(l+1)+14)+\beta((3l+1)\times k+6k+12k)$.

Once the cost function is decided, the objective is to minimize the cost function while meeting the inequalities defined by the security strength above. The optimal variables can be found by searching the variable space. Table II presents the optimal variables when α and β are 0.5. Without specification, we use it as the default setting in the following presented experiments. For comparison, we also list the recommended key sizes of Paillier, which were applied in a referred paper for comparison [27].

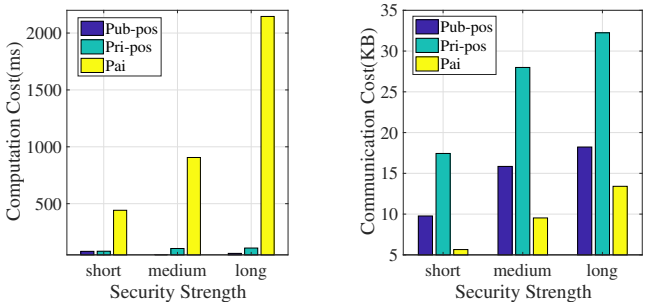


Fig. 3: Computation and communication comparison of different protocols

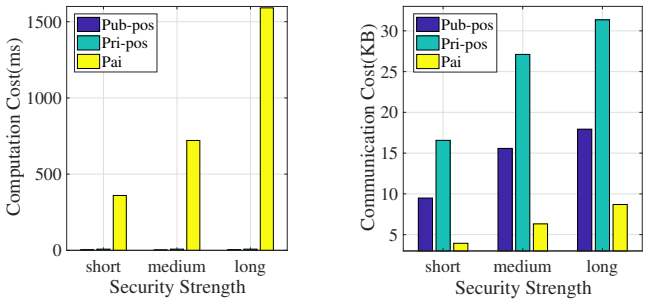


Fig. 4: Computation and communication cost of target point in different protocols

D. Discussion of Practical Usage

Due to the influence of noise, there is an error between a measured distance and the real distance, which results in deviation of positioning results or even failure of positioning calculation. The proposed protocols are theoretically accurate for trilateration positioning on clean data. However, the influence of noise data is an unavoidable issue in practice.

This problem can be solved by integrating effective noise reduction methods with our proposed protocols. For example, Yan et al. [28] proposed a method to enhance the reliability of measured information by deducting environmental noise with a model trained from historical measurement data. Li et al.

[26] proposed a method to estimate a position precisely by applying two machine learning models for position calculation with attack detection and tracing. By integrating these methods with our proposed protocols, it is possible to achieve highly accurate positioning with privacy preservation.

VII. Performance Evaluation

In this section, we evaluate the performance of the proposed protocols and compare them with Paillier based solution denoted as Pai, which is presented by Jiang et al. in [14]. Pai assumes the public availability of reference points location and the privacy of measured distance and computed results. In [14], based on the public availability of reference points location, A^{-1} can be calculated by the computation server easily. Also, the homomorphic addition property of Paillier makes it possible to calculate $A^{-1}B$ based on the encrypted B, which contains the sensitive distance information. However, Pai can only solve the positioning based on public reference points. For positioning based on private points, the challenge also resides in the computation of A^{-1}, since Paillier is unable to support the division and multiplication operations. Pri-pos is the first work solving the privacy protected positioning on private reference points. Therefore, we only compare the performance between Pub-pos and Pai.

A. Experimental Setup

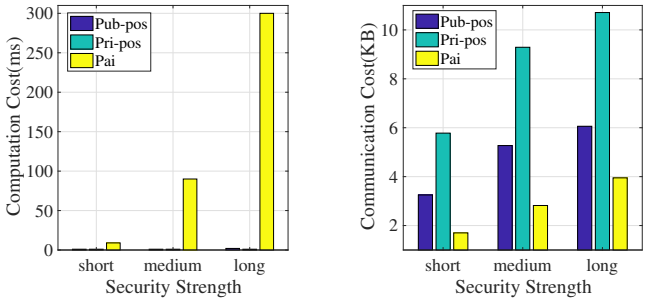
We implement the protocols in Java on a macOS platform with a 3.1GHz Intel Core i5 CPU and 8G RAM. The target point, the reference points and the computation server are implemented with three different entities and communicating through a client-server socket. We repeat the experiment 10 times and report average results. In each experiment, we generate five random positions and link them to the target point and the reference points randomly. The distances between the target point and the reference points are calculated based on the position information and owned by the reference points. Based on this setting, we implement our protocols for privacy-preserving positioning. Specifically, the reference points send the position and distance information to the computation server according to our protocol. The computation server processes the information and returns the estimated position to the target point. During the experiments, the computation and communication costs of each entity are measured based on its computation time and transferred data size, respectively. We record and compare the computation and communication cost of different protocols under different security strength. The optimal variables of each security level are decided according to the discussion results in Table II.

B. Performance of Positioning System

Fig. 3 shows the performance of protocols during the whole positioning process. Fig. 3a records the processing time of positioning in each protocol from position query to responded result received. Fig. 3b records the transferred data between the target point, the reference points and the computation server. From the figure, we can see that (1) Our proposed protocol Pub-pos and Pri-pos are at least 4.5 times faster than Pai in terms of positioning service provision. The running time of our protocol is always under 100ms while Pai takes 450ms for running "short" level security and 2200ms for "long" level security. (2) Pub-pos and Pri-pos present a stable service latency even under different security strengths while Pai's latency increases exponentially to security strength. (3) Even the communication cost of our proposed protocols are higher than Pai, they are still in an acceptable range as the communication cost for the whole system is no more than 35 KB. (4) The communication cost of all protocols increases with the increase of security strength. From observation, we can conclude that our proposed protocols are much more efficient and stable than Pai regarding positioning service provision (i.e., service latency) despite a slight increase in communication cost.

C. Performance of Target Point

Fig. 4 shows the performance of the target point in each protocol. Fig. 4a records the time that the target point spends in generating encryption key and decryption results and Fig. 4b records the transferred data of the target point, which include the keys shared with the reference points and the received data from the computation server. From the figure, we can see that (1) The computation time of the target point in Pub-pos and Pri-pos is much less than that in Pai. (2) The computation time of the target point remains stable in Pub-pos and Pri-pos while increasing linearly in Pai with the increase of security strength. (3) The communication cost of target points in Pub-pos and Pri-pos is higher than that in Pai but under 32 KB. (4) By comparing Fig. 4 and Fig. 3, it is easy to find that the target point makes up the most computation and communication cost in the whole system.

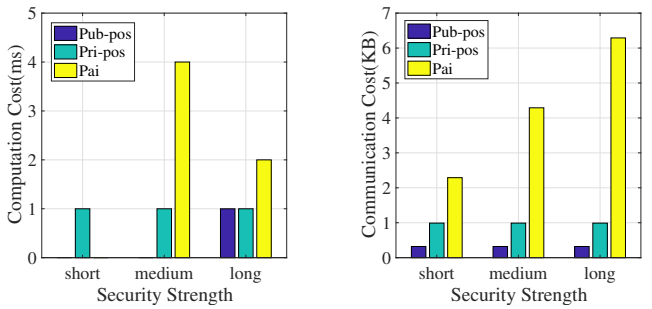


(a) (b)

Fig. 5: Computation and communication cost of reference point in different protocols

D. Performance of Reference Point

Fig. 5 shows the performance of the reference point in each protocol. Fig. 5a records the time spent in encrypting its data matrix at the reference point and Fig. 5b records the transferred data of the reference point, which include the key received from the target point and the encrypted matrix sent to the computation server. From the figure, we can see that (1) The



(a) (b)

Fig. 6: Computation and communication cost of computation server in different protocols

computation time of the reference point in Pub-pos and Pri-pos is much less than that in Pai. (2) The computation time of all of the protocols increases with the increase of security strength. (3) The communication cost of the reference points in Pub-pos and Pri-pos is higher than that in Pai, but all of them are under 11 KB. (4) The communication cost of the reference points in all of the protocols increases with the increase of security strength;

E. Performance of Computation Server

Fig. 6 shows the performance of the computation server in each protocol. Fig. 6a records the time spent by it in computation and Fig. 6b records its transferred data, which include the data received from the reference points and data sent to the target point. From the figure, we can see that (1) Both the computation cost and communication cost of the computation server in Pub-pos and Pri-pos are less than those in Pai. (2) The computation time of the computation server in both of the proposed protocols are quite stable even under different security strength. (3) The communication costa of all the protocols increase with security strength increasing, but all of them are less than 7 KB;

In short, the computation costs of our proposed protocols are much lower than the solution based on Paillier although with a slight increase in communication cost. Besides, the stability of service latency in terms of different security strength show a specific advantage of our protocol in providing high-security guarantee under the same constraints of service provision time.

TABLE III: Position Accuracy with Noise Influence

Methods	Accuracy
Posc	0.52m
Posn	1.48m
Yan-Posn [28]	0.86m
Li-Posn [26]	0.92m

F. Performance under Noise Influence

To simulate the real world, we introduce noise into the measured distance d. The noise is a random value generated from the normal distribution with mean=0 and standard derivation=1. The generated random value is added to distance d. We integrate the noise reduction methods proposed by Yan et al. [28] and Li et al. [26] into our protocols. The influence is measured with positioning accuracy, which is the mean deviation of the estimated position to the true position.

Table III shows the positioning accuracy by applying four methods. Posc stands for a positioning method that uses clean data without any noise. Posn refers to the positioning method that does not apply any noise reduction methods. While Yan-Posn and Li-Posn stand for the methods that apply the noise reduction methods proposed by Yan et al. [28] and Li et al. [26], respectively.

By comparing Posc and Posn, we can see that the positioning accuracy is sensitively impacted by noise, which is decreased from 0.52 meter to 1.48 meter. To improve it, we implement Yan-Posn and Li-Posn. From the results, we can see that both of them can improve the positioning accuracy effectively and Yan-Posn presents a better result than Li-Posn. This experiment shows that the influence of noise can be removed effectively by applying noise reduction methods in our proposed protocols.

VIII. Conclusion and Future Work

This paper proposed two protocols to overcome the privacy concern in 5G enabled positioning systems. By encrypting the original data matrix with two random matrices through matrix concatenation and multiplication, we can protect the private data of reference points while keeping the positioning service intact. We analyzed the security strength and costs of the proposed two protocols with optimal variable selection. We measured the performance of our protocols and compared it with a Paillier-based solution in terms of system performance and the performance of each participant, focusing on computation cost and communication cost, as well as aggregated overhead. The result shows that our proposed protocols present a better performance in both system aspects and individual aspects. Besides, the performance stability of our protocols under different security strengths outperforms, thus they can provide higher security guarantees under quantified time and communication constraints.

The proposed protocols are designed based on trilateration methods for positioning scenarios, but it can also be applied into other privacy-preserving scenarios that can be concluded as quadratic equation problems, such as outsourced quadratic equation verification. In our future work, we plan to adapt the two protocols for outsourced computation verification. We also plan to implement the prototype and evaluate the protocols performance on the real-world environment.

Acknowledgment

The work is supported in part by the Academy of Finland under Grants 308087, 335262, 345072 and 350464, the National Natural Science Foundation of China under Grants 62072351, the open research project of ZheJiang Lab under grant 2021PD0AB01, the Shaanxi Innovation Team Project under Grant 2018TD-007, and the 111 project under grant B16037.

REFERENCES

[1] Y. Lu, M. Koivisto, J. Talvitie, E. Rastorgueva-Foi, M. Valkama, and E. S. Lohan, "Cooperative positioning system for industrial IoT via mmwave Device-to-Device communications," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. IEEE, 2021, pp. 1–7.

[2] N. Chukhno, S. Trilles, J. Torres-Sospedra, A. Iera, and G. Araniti, "D2D-based cooperative positioning paradigm for future wireless systems: A survey," *IEEE Sensors Journal*, 2021.

[3] X. Cui, T. A. Gulliver, H. Song, and J. Li, "Real-time positioning based on millimeter wave device to device communications," *IEEE Access*, vol. 4, pp. 5520–5530, 2016.

[4] A. Kakkavas, M. H. C. Garcia, R. A. Stirling-Gallacher, and J. A. Nossek, "Multi-array 5G V2V relative positioning: Performance bounds," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 206–212.

[5] Z. Li, X. Zhao, Z. Zhaoa, and T. Braun, "WiFi-RITA positioning: Enhanced crowdsourcing positioning based on massive noisy user traces," *IEEE transactions on wireless communications*, 2021.

[6] Z. Wang, R. Liu, Q. Liu, J. S. Thompson, and M. Kadoch, "Energy-efficient data collection and device positioning in UAV-assisted IoT," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1122–1139, 2019.

[7] Q. Pei, B. Kang, L. Zhang, K.-K. R. Choo, Y. Zhang, and Y. Sun, "Secure and privacy-preserving 3D vehicle positioning schemes for vehicular ad hoc network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, pp. 1–12, 2018.

[8] S. Capkun, S. Ganeriwal, F. Anjum, and M. Srivastava, "Secure RSS-based localization in sensor networks," *Technical Report/ETH Zurich, Department of Computer Science*, vol. 529, 2011.

[9] S. H. Lee and J. Civera, "Triangulation: Why optimize?" *arXiv preprint arXiv:1907.11917*, 2019.

[10] F. Benassi, E. DallAsta, F. Diotri, G. Forlani, U. Morra di Cella, R. Roncella, and M. Santise, "Testing accuracy and repeatability of UAV blocks oriented with GNSS-supported aerial triangulation," *Remote Sensing*, vol. 9, no. 2, p. 172, 2017.

[11] A. R. Benjamin, D. OBrien, G. Barnes, B. E. Wilkinson, and W. Volkmann, "Improving data acquisition efficiency: Systematic accuracy evaluation of GNSS-assisted aerial triangulation in UAS operations," *Journal of Surveying Engineering*, vol. 146, no. 1, p. 05019006, 2020.

[12] S. U. Hussain and F. Koushanfar, "P3: Privacy preserving positioning for smart automotive systems," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 23, no. 6, pp. 1–19, 2018.

[13] T. Shu, Y. Chen, J. Yang, and A. Williams, "Multi-lateral privacy-preserving localization in pervasive environments," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 2319–2327.

[14] H. Jiang, H. Wang, Z. Zheng, and Q. Xu, "Privacy preserved wireless sensor location protocols based on mobile edge computing," *Computers & Security*, vol. 84, pp. 393–401, 2019.

[15] A. Bogdanov, E. Maneva, and S. Riesenfeld, "Power-aware base station positioning for sensor networks," in *IEEE INFOCOM 2004*, vol. 1. IEEE, 2004.

[16] M.-F. Tsai, P.-C. Wang, C.-K. Shieh, W.-S. Hwang, N. Chilamkurti, S. Rho, and Y. S. Lee, "Improving positioning accuracy for VANET in real city environments," *The Journal of Supercomputing*, vol. 71, no. 6, pp. 1975–1995, 2015.

[17] S. Liu and Z. Yan, "Verifiable edge computing for indoor positioning," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[18] S. Liu, Z. Yan, and R. Kantola, "Privacy-preserving D2D cooperative location verification," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021.

[19] Z. Yan, X. Qian, S. Liu, and R. H. Deng, "Privacy protection in 5G positioning and location-based services based on SGX," *ACM Transactions on Sensor Networks (TOSN)*, 2021.

[20] A. Konstantinidis, G. Chatzimilioudis, D. Zeinalipour-Yazti, P. Mpeis, N. Pelekis, and Y. Theodoridis, "Privacy-preserving indoor localization on smartphones," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 11, pp. 3042–3055, 2015.

[21] A. M. Sazdar, N. Alikhani, S. A. Ghorashi, and A. Khonsari, "Privacy preserving in indoor fingerprint localization and radio map expansion," *Peer-to-Peer Networking and Applications*, vol. 14, no. 1, pp. 121–134, 2021.

[22] H. Li, L. Sun, H. Zhu, X. Lu, and X. Cheng, "Achieving privacy preservation in WiFi fingerprint-based localization," in *Ieee Infocom 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 2337–2345.

[23] Z. Yang and K. Järvinen, "The death and rebirth of privacy-preserving WiFi fingerprint localization with Paillier encryption," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1223–1231.

[24] L. Schauer, F. Dorfmeister, and F. Wirth, "Analyzing passive Wi-Fi fingerprinting for privacy-preserving indoor-positioning," in *2016 International Conference on Localization and GNSS (ICL-GNSS)*. Ieee, 2016, pp. 1–6.

[25] P. Speciale, J. L. Schonberger, S. B. Kang, S. N. Sinha, and M. Pollefeys, "Privacy preserving image-based localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5493–5503.

[26] Y. Li, S. Liu, Z. Yan, and R. H. Deng, "Secure 5G positioning with truth discovery, attack detection and tracing," *IEEE Internet of Things Journal*, 2021.

[27] E. Barker, E. Barker, W. Burr, W. Polk, M. Smid *et al.*, *Recommendation for key management: Part 1: General*. National Institute of Standards and Technology, Technology Administration, 2006.

[28] M. Yan, F. Xu, S. Bai, and Q. Wan, "A noise reduction fingerprint feature for indoor localization," in *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2018, pp. 1–6.

Shushu Liu received the B.Sc. and M.Sc. degrees in computer science from Soochow University, Suzhou, China, in 2014 and 2017, respectively. She is currently pursuing the Ph.D. degree with the Department of Communication and Networking, Aalto University, Espoo, Finland. Her research interests are in social network, machine learning, and data security and privacy.

Zheng Yan received the B.Eng. degree in electrical engineering and the M.Eng. degree in computer science and engineering from the Xian Jiaotong University, Xian, China, in 1994 and 1997, respectively, the second M.Eng. degree in information security from the National University of Singapore, Singapore in 2000, and the Licentiate of Science and the Doctor of Science in Technology in electrical engineering from Helsinki University of Technology, Helsinki, Finland in 2005 and 2007. She is currently a professor at the Xidian University, Xian, China and a visiting professor at the Aalto University, Espoo, Finland. She authored over 300 peer-reviewed publications and solely authored two books. 80+ of her invented patents have been adopted by industry. Her research interests are in trust, security and privacy, and data analytics. Prof. Yan served and is serving as an area/associate editor of Information Fusion, IEEE Internet of Things Journal, IEEE Network Magazine, IEEE Access Journal, Information Sciences, JNCA, Security and Communication Networks, etc. She served as a steering, organization and program committee chair for over 30 international conferences. She is a fellow of IET and a senior member of the IEEE.